

UNIVERSITAT POLITÈCNICA DE CATALUNYA
(UPC) – BARCELONATECH

FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)

Coink

Disseny i implementació d'una webApp pel control de
despeses personals i en grup



Autora: Carlota Catot Bragós

Especialitat: Enginyeria del Software

Director: Maria José Casañ (ESSI)

Data de la defensa: 2 de Juliol del 2019

Titulació: Grau en enginyeria informàtica

Resum

Un dels problemes de la nostra societat és el de l'estalvi. Avui en dia molta gent estalvia poc o gens, i aquest és un problema que a la llarga pot repercutir negativament tant a la vida de la gent com a l'economia del país. Aquesta manca d'estalvi pot venir donada per diversos factors, però un d'ells és el simple fet que generalment la gent no és conscient de les despeses que fa, i per aquest motiu s'ha plantejat la següent pregunta: **Com podríem ajudar a la gent a estalviar amb l'ajuda de les TIC?**

Com a treball de final de grau, es presenta Coink, una WebApp per tal d'ajudar als usuaris a poder controlar les seves despeses, tan individuals com formant part d'un grup. Durant aquest document es presenten les diferents funcionalitats que ofereix aquesta aplicació, no abans de realitzar tot l'estudi previ al desenvolupament i explicar amb detall les estratègies de desenvolupament i validació.

Resumen

Uno de los problemas de nuestra sociedad es el ahorro. Hoy en día mucha gente ahorra poco o nada, y este es un problema que a largo plazo puede repercutir de manera negativa tanto en la vida cotidiana de la gente, como en la economía del país. Esta falta de ahorro puede deberse a muchos factores, pero uno de ellos es el simple hecho de que generalmente la gente no es consciente de los gastos que realiza, y por este motivo se ha planteado la siguiente pregunta: **¿Como podríamos ayudar a la gente a ahorrar con la ayuda de las TIC?**

Como trabajo de final de grado se presenta Coink, una WebApp para ayudar a los usuarios a poder controlar sus gastos individuales o formando parte de un grupo. Durante este documento se presentan las diferentes funcionalidades que ofrece esta aplicación, no antes de realizar todo el estudio previo al desarrollo y explicar detalladamente las estrategias de desarrollo y validación.

Abstract

One of the main problems in our society is the saving. Nowadays, many people save little or nothing, and this is a problem than can, in the long run, adversely affect both people's lives and the country's economy. This lack of savings can be caused by several factors, one of them being the simple fact that, generally, people are not aware of the expenses they make, and for this reason the following question has been raised: **How could we help people save with the help of ICTs?**

As final degree project, we present Coink, a WebApp to help users to be able to control their expenses, both their individual and group ones. This document showcases the different functionalities offered by this application, preceeded by the entire pre-development study and the details about both the development and the validation strategies.

Agraïments

M'agradaria agrair aquest projecte i el significat d'aquest (acabar el grau) a diferents persones. En primer lloc, a la meva família que ha aguantat el meu teclat fent soroll fins altes hores de la nit per realitzar projectes de la carrera; però en especial a la meva mare que em va animar a estudiar-la i gaudir-la; també, dins la família vull incloure a tietes Xavi i Carla, cosins, avis i padrins. També m'agradaria agrair als companys que he tingut al llarg la carrera en especial al Carlos que m'ha ajudat a dissenyar el logo de l'aplicació i m'ha donat molt suport durant total els anys, a l'Andreu que m'ha ajudat en diversos problemes que he tingut amb les configuracions d'AWS i tots els que han testejat l'aplicació (Didac, Juan Miguel, Víctor, Bernat i a la resta d'amics). De l'entorn de la feina m'agradaria destacar al meu tutor de pràctiques, Quim Muntal, que m'ha donat molt suport en els obstacles d'implementació que he tingut durant el desenvolupament del sistema. Per últim, vull fer esment a tots els professors que m'han patit i als que he patit, als que m'han ensenyat i dels que he après, en especial a la meva directora del projecte, Maria José Casañ per la seva paciència.

Gràcies a tots i totes!

Índex

1	Introducció	5
2	Problema a resoldre	7
3	Actors implicats (Stakeholders)	9
3.1	Desenvolupadora del projecte	9
3.2	Directora del projecte	9
3.3	Beneficiaris	9
4	Estudi de mercat	11
4.1	Aplicacions analitzades	11
4.2	Splitwise	11
4.3	Tricount	12
4.4	Expense Manager - Money tracker	13
4.5	iSpending	13
4.6	Fintonic	14
4.7	Conclusions	15
5	Objectius	17
5.1	Objectiu principal	17
5.2	Subobjectius	17
5.2.1	Complir propietats de WebApp	17
5.2.2	Gestió de tràmits completa	17
5.2.3	Gestió d'Usuaris	17
5.2.4	Gestió de Grups	17
5.2.5	Bona arquitectura	17
5.3	Objectius pel desenvolupador del projecte	18
5.4	Possibles objectius posteriors al projecte	18
6	Abast	19
6.1	Metodologia i rigor	20
6.2	Possibles obstacles	21
6.2.1	Temps	21
6.2.2	Complexitat	21
7	Planificació Inicial	23
7.1	Descripció de tasques	23
7.2	Recursos	25
7.3	Estimació de temps	25
7.4	Diagrama de Gantt	26
7.5	Pla d'acció	27
8	Planificació final	29

9 Gestió econòmica	31
9.1 Costos Directes	31
9.1.1 Tasques inicials (GEP)	31
9.1.2 Desenvolupament Back-End	32
9.1.3 Desenvolupament Front-End	32
9.1.4 Millores	33
9.1.5 Documentació	33
9.1.6 Total costos directes	33
9.2 Costos Indirectes	34
9.3 Costos Imprevistos	34
9.4 Costos Totals	34
9.5 Control de pressupost	35
10 Cost real del projecte	37
10.1 Costos directes reals	37
10.1.1 Tasques inicials (GEP)	37
10.1.2 Desenvolupament Back-End	37
10.1.3 Desenvolupament Front-End	38
10.1.4 Millores	38
10.1.5 Documentació	38
10.2 Total costos directes	39
10.3 Costos Totals	39
11 Sostenibilitat i impacte social	41
11.1 Dimensió ambiental	41
11.2 Dimensió social	41
11.3 Dimensió Econòmica	42
12 Requisits	43
12.1 Requisits no funcionals	43
12.2 Requisits funcionals	44
12.3 Criteris d'acceptació	46
13 Disseny del sistema	51
13.1 Arquitectura de Software del sistema	52
13.2 Patrons de disseny	53
13.3 Esquema de les bases de dades	56
13.4 Disseny de les interfícies - Funcionalitats aplicació	58
13.4.1 Procés de login i logout	58
13.4.2 Procés de registre	59
13.4.3 Pantalla principal	62
13.4.4 Contactes	66
13.4.5 Grups	67
13.4.6 Perfil	69
14 Diari d'execució	71
14.1 Sprint 1	71
14.2 Sprint 2	73
14.3 Sprint 3	76
14.4 Sprint 4	80
14.5 Sprint 5	84
14.6 Sprint 6	89
14.7 Sprint 7	91
14.7.1 Resum d'hores invertides	93
14.8 Configuracions finals AWS	93
14.8.1 API Gateway	93
14.8.2 Lambdas	94
14.8.3 DynamoDB	95
14.8.4 Cognito	96
14.9 Release burndown and velocity chart	96

15 Estrategia de treball i desenvolupament	99
16 Estrategia de testing i validació	101
16.1 Testing AWS	101
16.2 Validació amb usuaris reals	102
17 Obstacles	105
17.1 Existència de coink.com	105
17.2 Temps	105
17.3 Funcionament Amazon Web Services	106
18 Conclusions	107
18.1 Assoliment d'objectius del projecte	107
18.2 Assoliment competències tècniques del projecte	107
18.3 Treball futur	108
18.4 Conclusions personals	109
Bibliografia	I
Índex de figures	IV
Índex de taules	VI

Capítol 1

Introducció

Un dels problemes de la nostra societat és el de l'estalvi. Avui en dia molta gent estalvia poc o gens. Per exemple, a l'estudi realitzat pel diari l'Economista [1] l'octubre de 2018, es parla de què a Espanya s'estalvia la meitat que al conjunt de la zona euro. Això ho podem observar a la imatge 1.1, on es veu com a partir del 2010 la taxa d'estalvi ha anat decreixent a Espanya, mentre que a la zona euro s'ha mantingut estable, tot i això podem veure que el percentatge d'estalvi més alt és el d'Alemanya amb un 17,3%, que tampoc es pot considerar un rang molt alt. Aquest és un problema que a la llarga pot repercutir negativament tant a la vida de la gent com a l'economia del país. Aquesta manca d'estalvi pot venir donada per diversos factors, però un d'ells és el simple fet que generalment la gent no és conscient de les despeses que fa, i per aquest motiu s'ha plantejat la següent pregunta: **Com podríem ajudar a la gent a estalviar amb l'ajuda de les TIC?**

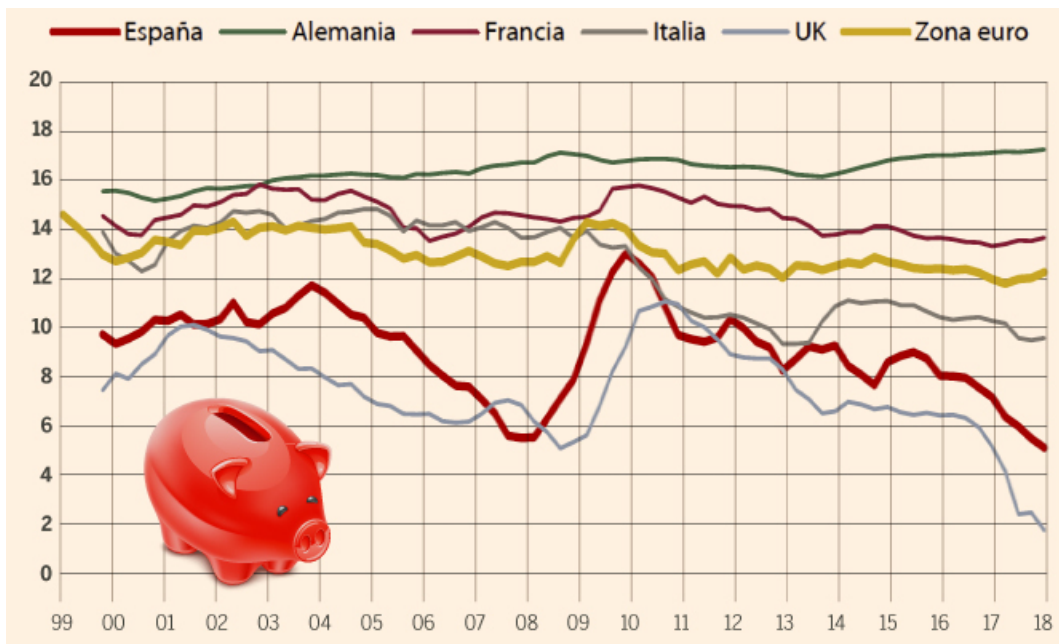


Figura 1.1: Taxa d'estalvi a les llars (percentatge de renda brut disponible) [2]

Capítol 2

Problema a resoldre

Com s'ha comentat prèviament a la introducció, a la societat occidental és molt comú tenir poc control de les despeses del dia a dia, això fa que a poc a poc els diners s'acabin sense avisar i fan difícil l'estalvi. Per tal de poder respondre a la pregunta plantejada a la introducció: *Com podríem ajudar a la gent a estalviar?*, s'ha fet una cerca de les solucions TIC proposades fins al moment (presentades a l'estudi de mercat) i s'ha plantejat una solució que més endavant s'ha dissenyat i desenvolupat.

Com a treball de final de grau, es presenta una WebApp per tal d'ajudar als usuaris a poder controlar les seves despeses i d'aquesta manera puguin estalviar. Per tal que l'usuari s'interessi en el nostre sistema caldrà realitzar una aplicació atractiva, compatible per diferents dispositius, i que tingui funcionalitats bàsiques i usables.

Capítol 3

Actors implicats (Stakeholders)

3.1 Desenvolupadora del projecte

Aquest projecte tindrà una única desenvolupadora, aquesta serà l'encarregada de la realització de totes les tasques associades al projecte, el desenvolupament del codi, els testos del propi i la documentació requerida. A més haurà de seguir els períodes d'entrega, l'horari definit i la planificació acordada amb la directora del projecte per tal de poder completar el projecte en el termini establert. La desenvolupadora del projecte serà la Carlota Catot Bragós, estudiant cursant l'últim quadrimestre del grau d'enginyeria informàtica en especialització en enginyeria del software a la Facultat d'Informàtica de Barcelona (UPC).

3.2 Directora del projecte

La directora del projecte serà l'encarregada de guiar i ajudar a la desenvolupadora del projecte en els possibles obstacles que es pugui trobi durant el desenvolupament. La directora d'aquest projecte és la Maria José Casañ, professora del departament d'ESSI de la Facultat d'Informàtica de Barcelona (UPC).

3.3 Beneficiaris

Per últim tenim els beneficiaris, aquest projecte consta de beneficiaris directes i indirectes, com a beneficiaris directes podem trobar a qualsevol usuari que estigui donat d'alta al sistema, aquest serà el que està interessat amb estalviar i per tant s'ha creat un compte i és usuari actiu per poder controlar les seves despeses i formar part de la comunitat. Per altra banda, com a usuaris indirectes trobarem aquells usuaris que no estan donats d'alta al sistema, però que estan afectades per l'ús de l'aplicació d'una manera indirecta, ja sigui perquè l'usa un familiar amb qui comparteix despeses o perquè els amics l'usen per dividir despeses compartides.

Capítol 4

Estudi de mercat

Per realitzar l'estudi de mercat s'ha fet una cerca d'aplicacions mòbils que es poden trobar a Google Play (Android), a l'App Store (iOS), via web o la combinació d'algunes de les possibilitats anteriors. Durant la cerca es van trobar moltes aplicacions que tenien funcionalitats semblants, i per tal de fer un estudi de mercat més acotat és s'ha optat per seleccionar 5 aplicacions amb certa semblança i diferents funcionalitats per tal de cobrir més possibilitats dins el mercat.

Per totes les aplicacions es va seguir el mateix procediment, per tal de trobar les funcionalitats bàsiques d'aquestes es van descarregar i provar durant una setmana per tal de trobar les funcionalitats genèriques més semblants entre elles. Posteriorment es presentarà una conclusió de l'estudi de mercat per tal de poder buscar una solució que pugui incloure les funcionalitats més destacades o les que es consideren determinants a partir de l'estudi realitzat.

4.1 Aplicacions analitzades

4.2 Splitwise

Splitwise és una aplicació (amb versió mòbil [3] i web [4]) que ofereix poder dividir despeses en grup o amb contactes específics per tal de repartir les despeses que s'han realitzat en un grup on algú ha pagat per la resta del grup o una part d'aquest. Les funcionalitats a destacar d'aquesta aplicació són les següents:



- #1 - Tràmits: Es poden afegir, esborrar i editar les despeses. S'afegeixen per un membre del grup que haurà d'indicar qui ha fet el pagament i entre quins usuaris s'ha de dividir, aquest es pot separar de manera equitativa o de manera desigual. No és necessari que tots els membres del grup participin en una despesa.
- #2 - Entrada Manual: Els tràmits s'han d'afegir manualment per un dels membres del grup, no és necessari que aquest participi a la despesa.
- #4 - Saldo: L'aplicació compta amb una secció on s'informa l'usuari de què deu i el que se li deu (sumant les despeses que hi ha a diferents grups), i una suma dels dos oferint el saldo total. El saldo es pot veure per grup o el general.
- #5 - Perfil: Per tal d'utilitzar l'aplicació cal tenir un perfil vinculat a un número de telèfon o a un correu electrònic, d'aquesta manera es podrà accedir al compte tant des de l'aplicació mòbil com des de la web. El perfil es podrà gestionar per canviar les dades en qualsevol moment.
- #6 - Notificacions: L'usuari rep una notificació en el cas que sigui afegit a un grup o se li reclamin diners. Les notificacions es poden desactivar en cas de no voler-ne rebre.
- #7 - Contactes: Un usuari pot afegir contactes, a través del número de telèfon o del correu electrònic. Es pot reclamar diners a un contacte des de dins d'un grup o des del llistat de contactes de manera individual.

- #8 - Grups: Un usuari pot formar part de diferents grups. Aquests es poden crear a partir de la llista de contactes o pel número de telèfon o correu electrònic (en cas de no tenir a l'usuari al llistat de contactes), la resta d'usuaris del grup (que no l'han creat) s'uneixen al grup rebent una invitació. Els grups es poden editar o eliminar en qualsevol moment.
- #9 - Balanç de grup: Per cada grup, l'usuari pot consultar el que cada membre del grup deu a la resta.
- #10 - Multidivisa: L'aplicació ofereix poder canviar de divisa en qualsevol moment, de manera que es poden barrejar divises i, per exemple, pot posar que deus 10€ i 10\$, però no fa la conversió per tal d'unificar en una sola moneda.
- #12 - Activitat contactes: L'aplicació compta amb una secció que permet veure un resum de tots els moviments realitzats per tots els contactes dins dels grups als quals l'usuari pertany.

4.3 Tricount

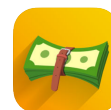
Tricount és una aplicació (amb versió mòbil[5] i web[6]) molt semblant a Splitwise. Aquesta també ofereix poder dividir despeses en grups, amb algunes diferències destacables, com per exemple no cal tenir un perfil per poder formar part d'un grup. Les funcionalitats a destacar d'aquesta aplicació són les següents:



- #1 - Tràmits: Es poden afegir, esborrar i editar les despeses. S'afegeixen per un membre del grup que haurà d'indicar qui ha fet el pagament i entre quins usuaris s'ha de dividir, aquest es pot separar de manera equitativa o de manera desigual. No és necessari que tots els membres del grup participin en una despesa.
- #2 - Entrada Manual: Els tràmits s'han d'afegir manualment per un dels membres del grup, no és necessari que aquest participi a la despesa.
- #4 - Saldo: L'aplicació ofereix, per cada grup, un resum de què s'ha gastat el grup en general i també el que s'ha gastat l'usuari que està consultant l'aplicació.
- #5 - Perfil*: Per tal d'utilitzar l'aplicació no és necessari que l'usuari disposi d'un perfil, però hi ha l'opció de poder tenir un perfil, aquest es pot crear mitjançant un correu electrònic, o vinculant amb Facebook o Google.
- #8 - Grups: Un usuari pot pertànyer a diferents grups, aquests es creen entrant el nom dels usuaris que han de formar part del grup, per poder entrar al grup cal rebre una invitació i indicar quin és el teu usuari per tal de vincular la conta amb aquell membre del grup. Els grups es poden editar o eliminar en qualsevol moment.
- #9 - Balanç de grup: Per cada grup, l'usuari pot consultar el que cada membre del grup deu a la resta.
- #13 - Pagar: Si l'usuari està registrat i per tant l'aplicació sap quines són les seves despeses, aquest pot pagar-les a través de PayPal.
- #14 - Compartir: Un grup es pot compartir a través de les diferents xarxes socials, d'aquesta manera es poden enviar invitacions a usuaris que no tenen una conta, i podran veure les diferents despeses realitzades i el total de què deuen a cadascú.
- #15 - Ordenar: Les despeses dins un mateix grup es poden ordenar per títol, quantitat, data o persona que ha pagat.
- #16 - Imatges: Un usuari pot afegir imatges a les despeses, per tal de poder adjuntar el tiquet o una imatge que tingui relació.

4.4 Expense Manager - Money tracker

Expense Manager és una aplicació (només mòbil[7]) que et permet tenir un control de les despeses realitzades al dia a dia, per tant, les dades s'introdueixen manualment, en introduir les dades manualment s'intueix que es controlaran sobretot les despeses en efectiu, ja que hi ha aplicacions (es veurà amb Fintonic) que estan destinades a controlar de manera automàtica les despeses realitzades mitjançant el banc. Les funcionalitats a destacar d'aquesta aplicació són les següents:



- #1 - Tràmits: Es poden afegir, esborrar i editar les despeses o els ingressos. Per cada tràmit cal indicar una quantitat, una categoria i opcionalment una nota informativa per tal de poder descriure la despesa.
- #2 - Entrada Manual: Els tràmits s'han d'afegir manualment.
- #4 - Saldo: L'aplicació ofereix una secció on es pot consultar el saldo mensual, veient les despeses, els ingressos i un total d'aquest. També es pot consultar únicament les despeses o únicament els ingressos de manera que estaran separats per categoria. Tota l'aplicació està basada en els mesos de l'any i és per això que només es pot veure el saldo mensual.
- #11 - Categories: Totes les despeses i ingressos estan categoritzats per tal de poder tenir el sistema més organitzat.
- #16 - Imatges: Un usuari pot afegir imatges a un tràmit (ingrés o despesa), per tal de poder adjuntar el tiquet, la nòmina o una imatge que tingui relació.
- #17 - Calendari: L'aplicació té com a pàgina principal un calendari, tots els tràmits es troben guardats el dia que es van realitzar per tant, per poder consultar tràmits concrets l'usuari ha d'anar al dia que es va realitzar.
- #18 - Gràfics: L'aplicació també ofereix gràfiques amb les seccions descrites a la funcionalitat de Saldo (pels ingressos, les despeses o el total del saldo mensual).

4.5 iSpending

iSpending és una aplicació (només mòbil[8] i únicament disponible per iOS) bastant semblant a l'aplicació anterior (Expense Manager) permet tenir control de les despeses realitzades introduint les dades manualment, a continuació amb les funcionalitats a destacar d'aquesta aplicació es podrà veure les diferències que té amb Expense Manager:



- #1 - Tràmits: Es poden afegir, esborrar i editar les despeses o els ingressos. Per cada tràmit cal indicar una quantitat, una categoria i opcionalment una nota informativa per tal de poder descriure la despesa.
- #2 - Entrada Manual: Els tràmits s'han d'afegir manualment.
- #4 - Saldo: L'aplicació ofereix una secció on es presenta la suma d'ingressos, de despeses i un total del saldo (la resta dels ingressos i les despeses), això es pot consultar per períodes de temps (un any, 3 mesos, 6 mesos, un mes o el dia actual).
- #11 - Categories: Totes les despeses i ingressos estan categoritzats per tal de poder tenir el sistema més organitzat.
- #17 - Calendari: L'aplicació no ofereix un calendari pròpiament, com hem vist a l'aplicació anterior, sinó que el resum dels tràmits fets es poden consultar en diferents períodes de temps: avui, una setmana, un mes, un trimestre o un any marcat pel dia actual.
- #18 - Gràfics: L'aplicació ofereix poder consultar gràfiques dels ingressos, les despeses i els estalvis (saldo actual) pels diferents períodes de temps definits a la funcionalitat anterior.

- #19 - Exportar a Excel: L'aplicació ofereix poder enviar per correu electrònic, un Excel amb tots els tràmits realitzats.
- #20 - Cerca: L'aplicació ofereix una secció on es permet buscar tràmits per nom, quantitat, nota descriptiva, categoria o qualsevol dels anteriors, en cas que no sàpigues que estàs buscant.
- #21 - Informe per període de temps: Es pot veure un resum en forma d'informe de diferents períodes de temps (1 mes, 3 mesos, 6 mesos i 1 any), en aquest apartat no es parteix de la data actual però es poden consultar períodes anteriors.

4.6 Fintonic

Fintonic és una aplicació (amb versió mòbil[9] i web[10]) que a diferència de les dues aplicacions anteriors, ofereix poder tenir un control de les despeses personals realitzades mitjançant una targeta de crèdit/dèbit d'una manera automàtica, ja que l'aplicació agafa les dades directament del compte bancari. Les funcionalitats a destacar d'aquesta aplicació són les següents:



- #3 - Entrada Automàtica: Tota la informació sobre els tràmits que té l'aplicació, ve donada de manera automàtica per les dades bancàries connectades des d'un inici per l'usuari en registrar la conta.
- #4 - Saldo: L'aplicació mostra el total del saldo disponible a tots els comptes bancaris vinculats al sistema (per separat).
- #5 - Perfil: Per tal d'utilitzar l'aplicació és necessari que l'usuari s'hagi registrat de manera que tingui un perfil vinculat a un correu electrònic i amb les dades bancàries dels diferents comptes que vol anar consultant.
- #6 - Notificacions: L'usuari pot rebre notificacions per diferents operacions realitzades al compte bancari i aquestes es poden gestionar en qualsevol moment.
- #14 - Categories: Tots els tràmits tenen assignats una categoria, aquesta s'assigna en funció del destinatari (en cas de despesa) o el remitent (en cas d'ingrés) del tràmit. Les categories es poden editar.
- #19 - Exportar a Excel: L'usuari pot exportar en Excel (també com a informe econòmic) per període de temps, i dels diferents comptes bancaris vinculats a l'usuari, el document exportat s'envia per correu electrònic.
- #20 - Cerca: L'aplicació ofereix a l'usuari poder cercar tràmits per concepte, comerç, quantitat...
- #21 - Informe per període de temps: L'aplicació ofereix informes resumits per categories i per període de temps, aquest pot ser mensual, setmanal, trimestral, anual o en un rang de dates personalitzat.
- #22 - Connexió amb bancs: Per tal de fer l'automatització del sistema, hi ha una connexió amb els bancs al qual s'accedeix amb les credencials bancàries de l'usuari, demanades al registre de l'usuari, tot i que es poden afegir nous comptes en qualsevol moment.
- #23 - Divisió càrrec en més d'una despesa: L'usuari pot dividir una mateixa despesa en diferents despeses per tal d'indicar diferents compres en un mateix pagament.

4.7 Conclusions

Per tal d'extreure una conclusió a l'estudi de mercat, s'ha realitzat la taula 4.1 amb totes les funcionalitats trobades a les aplicacions anteriors, que com s'ha pogut comprovar anteriorment a l'anàlisi de les diferents aplicacions s'han anat numerant i posat un nom genèric a cada funcionalitat encara que aquestes no siguin 100% iguals per tal de poder veure de manera gràfica i a grans trets les principals diferències entre les 5 aplicacions.

En analitzar la taula, podem veure que les aplicacions estan focalitzades a oferir només un tipus de servei, despeses en grup (en el cas de Splitwise i Tricount), control de despeses en manual (en el cas d'Expense Management i iSpending) o control de despeses realitzades a través del banc (en el cas de Fintonic). Tenint en compte això, es vol proposar una solució que tingui funcionalitats transversals per tal d'oferir diversitat i poder utilitzar una aplicació en lloc de 2 o 3 per la mateixa finalitat, controlar les despeses.

	Splitwise	Tricount	EManager	iSpending	Fintonic
#1 - Tràmits					
#2 - Entrada Manual					
#3 - Entrada Automàtica					
#4 - Saldo					
#5 - Perfil					
#6 - Notificacions					
#7 - Contactes					
#8 - Grups					
#9 - Balanç grupal					
#10 - Multidivisa					
#11 - Categories					
#12 - Activitat contactes					
#13 - Pagar					
#14 - Compartir					
#15 - Ordenar					
#16 - Imatges					
#17 - Calendari					
#18 - Gràfics					
#19 - Exportar a Excel					
#20 - Cerca					
#21 - Informe					
#22 - Connexió amb bancs					
#23 - Divisió càrrec					

Taula 4.1: Taula comparativa de les funcionalitats de les aplicacions a l'estudi de mercat

Capítol 5

Objectius

5.1 Objectiu principal

L'objectiu principal del projecte és dissenyar i desenvolupar una aplicació prou bona per a poder ajudar als usuaris d'aquesta a controlar les despeses realitzades i d'aquesta manera poder estalviar de manera més fàcil.

5.2 Subobjectius

Com a subobjectius es presenten algunes de les funcionalitats bàsiques i propietats que ha de complir l'aplicació per tal de complir els requeriments que es plantejaran més endavant en forma d'històries d'usuari, es destaquen com a subobjectiu el que es considera indispensable per l'aplicació.

5.2.1 Complir propietats de WebApp

Com a primer subobjectiu definit es vol realitzar una WebApp i per tant és necessari que l'aplicació sigui usable tant a un dispositiu mòbil com a un ordinador de qualsevol mida, en altres paraules, caldrà que l'aplicació sigui responsive.

5.2.2 Gestió de tràmits completa

Com a segon subobjectiu, per tal de presentar una aplicació completa de control de despeses i ingressos serà necessari tenir una gestió completa dels tràmits podent oferir classificació de tràmits pels camps especificats posteriorment als requisits.

5.2.3 Gestió d'Usuaris

Com a tercer subobjectiu, per tal que cada usuari pugui tenir el seu perfil i les seves dades personals, serà necessari tenir una gestió d'usuaris, sense aquest subobjectiu no es podran completar els subobjectius 2 i 3.

5.2.4 Gestió de Grups

Com a quart subobjectiu, per tal d'oferir un sistema complet serà necessari tenir grup i per tal de complir aquest subobjectiu serà necessari tenir una connectivitat entre els usuaris per tal que aquests puguin compartir despeses.

5.2.5 Bona arquitectura

Com a últim subobjectiu, per tal de tenir un bon sistema, es vol tenir una bona arquitectura i per tal de complir aquest subobjectiu serà necessari assolir els coneixements necessaris d'AWS (Amazon Web Services) per tal de tenir una bona arquitectura a la part de l'API (tot això està explicat al capítol de disseny del sistema).

5.3 Objectius pel desenvolupador del projecte

Per altra banda, com és un projecte de final de grau d'enginyeria informàtica un objectiu secundari serà l'aprenentatge de la desenvolupadora (i estudiant) de les diferents tecnologies i metodologies que s'usaran, Golang com a nou llenguatge de desenvolupament per la realització del Back-End i com a punt molt important, realitzar un bon Front-End per tal que sigui usable des del mòbil i des de l'ordinador.

5.4 Possibles objectius posteriors al projecte

Com a futurs objectius del projecte es podria millorar l'aplicació de moltes maneres afegint algunes de les característiques que s'ha marcat com a *OUT*, en concret es podria afegir integració amb els bancs per tal de poder automatitzar una mica el procés, ja que actualment és tot completament manual.

Capítol 6

Abast

Proposta Inicial

Per tal de definir l'abast del projecte podem partir del comentat als punts anteriors en concret podem centrarnos amb el comentat previament al problema a resoldre, a les conclusions de l'estudi de mercat i als objectius.

Es proposa dissenyar i desenvolupar una WebApp per tal que l'usuari pugui tenir un control de despeses personals i l'ajudi a estalviar. Com a punt important definit com al primer subobjectiu es que l'aplicació compleixi les característiques d'una WebApp, i la principal es que sigui usable per tots els dispositius encara que les vistes no siguin les mateixes per tots ells.

Per tal de començar s'ha definit un MVP (Minimum Viable Product) en forma de NOT list (taula 6.1) agafant les funcionalitats de les aplicacions analitzades a l'estudi de mercat tot i que algunes de les funcionalitats s'han canviat de nom per simplificar o s'han ajuntat amb alguna altra.

IN	OUT
<ul style="list-style-type: none">- Gestió d'usuaris- Grups- Contactes- Gestió de tràmits- Multiidioma- Categories- Balanç- Saldo- Cerca/Filtre/Ordenar	<ul style="list-style-type: none">- Connexió amb bancs- Aplicacions mòbils natives (iOS/Android)
TBD (To Be Determinate)	
<ul style="list-style-type: none">- Notificacions- Multidivisa- Chat- Exportar- Gràfics- Calendari- Compartir- Imatges- Informe- Divisió de despeses	

Taula 6.1: NOT list inicial del projecte

Resultat Final

Finalment, s'ha dissenyat i desenvolupat una WebApp per tal que l'usuari pugui tenir un control de despeses personals i l'ajudi a estalviar. S'ha realitzat tota l'aplicació amb una versió mòbil de la web per tal de complir les característiques de la WebApp, ja que d'aquesta manera l'aplicació és completament responsive.

A la taula 6.2 es presenta la NOT list un cop acabat el desenvolupament. De les funcionalitats que estaven per determinar, finalment s'han implementat les que podem veure en verd (Exportar, Gràfics i Informe). Pel que fa a les funcionalitats que s'ha decidit no implementar, es poden veure en vermell. Com a punt a destacar, hi ha més funcionalitats que han passat a l'OUT de les que han passat a l'IN, això ha estat degut al fet que des d'un inici es va decidir implementar les funcionalitats més ambicioses deixant per determinar funcionalitats extres que no són necessàries per complir els objectius del projecte.

IN	OUT
<ul style="list-style-type: none"> - Gestió d'usuaris - Grups - Contactes - Gestió de tràmits - Multiidioma - Categories - Balanç - Saldo - Cerca/Filtre/Ordenar - Divisió de despeses - Exportar - Gràfics - Informe 	<ul style="list-style-type: none"> - Connexió amb bancs - Aplicacions mòbils natives (iOS/Android) - Notificacions - Chat - Calendari - Compartir - Multidivisa - Imatges

Taula 6.2: NOT list final del projecte

6.1 Metodologia i rigor

Metodologia de treball

Durant el desenvolupament del projecte s'aplicaran tècniques pròpies de les metodologies Agile. Es treballarà en cicles curts (Sprints) i cadascun d'aquests estarà basat en un grup d'històries d'usuari definides a l'inici del projecte, aquestes històries seran dinàmiques, per tant, poden anar canviant durant el desenvolupament del projecte, tot i que les històries de cada Sprint s'hauran de definir a l'inici d'aquest. Després de cada Sprint s'extrauran conclusions del mètode de treball que s'ha seguit i l'eficiència d'aquest per tal de poder aconseguir més eficiència i rendiment durant els següents Sprints i d'aquesta manera poder assolir els objectius del projecte.

També s'organitzaran reunions amb la directora del projecte per poder obtenir un feedback extern i d'aquesta manera també facilitar la resolució dels possibles obstacles i dificultats que puguin aparèixer durant el desenvolupament.

Als capítols de diari d'execució i estratègia de treball i desenvolupament s'explica més en detall els diferents processos que ha tingut el treball i la manera de treballar que s'ha seguit.

Eines de seguiment

Per tal de poder fer un bon seguiment del projecte, tant com per saber el que queda per endavant, com el que s'ha realitzat fins al moment, i portar un bon control del desenvolupament, s'utilitzaran les següents eines:

- **GitHub:** S'utilitzarà per tenir el projecte a un repositori privat i així poder utilitzar el control de versions de Git, que permetrà tenir un seguiment de l'evolució del projecte amb les diferents versions d'aquest, i poder tornar enrere en cas de topiar amb un problema important.
- **Taiga:** S'utilitzarà per a poder organitzar la feina amb les diferents històries d'usuari durant els Sprints. Aquesta eina ens permetrà tenir un seguiment més exhaustiu del desenvolupament del projecte.

Metodologia de validació

Per tal de validar el codi es realitzaran testos unitaris en algunes de les parts, però la millor validació serà intentar l'ús en usuaris reals utilitzant a companys, amics i família per testejar. Els companys de carrera per tal que puguin trobar *bugs* al sistema, i amics i família per tal de provar l'experiència d'usuari.

Al capítol d'estratègia de testing i validació s'explica en detall els testos unitaris que s'han seguit i el feedback que s'ha rebut per part dels usuaris que han testejat l'aplicació.

6.2 Possibles obstacles

6.2.1 Temps

Com a possible obstacle es podria destacar el temps, ja que els objectius del projecte són bastant ambiciosos i el desenvolupament d'aquest s'ha de realitzar aproximadament quatre mesos, per tant, s'hauran de complir estrictament tots els períodes de temps presentats més endavant per tal que el projecte pugui estar completat a la data establerta.

6.2.2 Complexitat

Altrament, es pot considerar un possible obstacle el fet de voler desenvolupar funcionalitats complexes amb llenguatges i tecnologies dels quals la desenvolupadora del projecte no és del tot experta, i també que tota l'aplicació pugui ser multiplataforma, pot ser un coll d'ampolla, ja que no es té molt coneixement del tema i s'haurà d'aprendre durant el desenvolupament.

Capítol 7

Planificació Inicial

A continuació es descriurà la planificació prevista per aquest projecte. El temps estimat de duració és al voltant de 4 mesos. Es va començar el divendres 1 de febrer del 2019 i té com a data de finalització el 24 de juny del 2019, ja que la defensa del projecte serà la primera setmana de Juliol.

Cal destacar que els temps i la planificació presentada a continuació, s'haurà d'anar actualitzant a mesura que avança el desenvolupament del projecte, ja que en seguir una metodologia Agile, pot ser que apareguin noves tasques després de cada sprint o que canviïn les dates, tot i que s'intentarà ser fidel a la planificació programada.

7.1 Descripció de tasques

El projecte tindrà 5 seccions o fases es dividiran en diverses tasques, algunes d'elles es desenvoluparan de manera simultània, ja que no tenen precedència entre elles, sinó que es complementen:

Tasques Inicials (GEP)

En aquesta secció, es farà una visió general del projecte per definir, planificar, fer anàlisi de requisits i avaluar la viabilitat del projecte. Aquesta fase prendrà al voltant d'un mes i tindrà 4 tasques:

1. Definició de l'abast i contextualització
2. Planificació temporal
3. Gestió econòmica i sostenibilitat
4. Presentació final

Aquesta fase és molt important, ja que es definiran les directrius bàsiques del projecte, i un cop estan definides, es pot passar a les fases següents.

Desenvolupament Back-End

Durant aquesta fase es desenvoluparà el Back-End de l'aplicació, es superposa bastant en temps amb la fase definida a continuació *desenvolupament Front-End*, ja que són complementaries.

A partir d'aquesta fase es començarà a aplicar la metodologia *bugs* que consistirà en Sprints d'una setmana on s'agafaran les històries d'usuari que es volen desenvolupar durant aquesta fase i es treballaran.

La fase es dividirà en 6 grans tasques:

1. Anàlisi requisits

2. Disseny del sistema
3. Creació de la base de dades
4. Infraestructura Serverless
5. Desenvolupament de l'API
6. Tests - Validació

Aquestes tasques hauran de seguir aquest ordre tot i que algunes es superposaran en el temps (com es pot veure al Gantt que es presenta més endavant), això es farà així perquè no tenen precedència les unes amb les altres més aviat el contrari. Les dues primeres tasques sí que tenen precedència, és a dir, cal que facin abans de començar les tasques 3, 4 i 5, i la tasca 6 no es podrà acabar completament fins que la tasca 5 estigui acabada tot i que es pot començar, per tal no es considera que hi ha dependència.

Desenvolupament Front-End

Durant aquesta fase es desenvoluparà el Front-End de l'aplicació, i com s'ha comentat anteriorment, se superposa bastant en temps amb la fase definida anteriorment *desenvolupament Back-End*, ja que són complementaries.

Com per la fase anterior, durant aquesta fase s'aplicarà la metodologia *Agile*, que consistirà en Sprints d'una setmana on s'agafaran les històries d'usuari que es volen desenvolupar durant aquesta fase i es treballaran.

La fase es dividirà en 5 grans tasques:

1. Anàlisi requisits
2. Disseny del sistema
3. Desenvolupament de la lògica del sistema
4. Desenvolupament de les interfícies
5. Tests - Validació

Igual que a la fase anterior, les tasques hauran de seguir aquest ordre, ja que les tasques 1 i 2 precedeixen a les tasques 3 i 4 per tal de començar a desenvolupar primer s'ha de saber quins requisits s'han de seguir i com ha de ser el sistema. Amb el punt 5 passa com a la fase anterior, no es considera dependent de la 3 i la 4 però no es pot acabar sense que estiguin acabades.

Millores

Un cop acabat tot el desenvolupament (cap a finals de Juny) entrarem a la fase de les millores on es realitzaran 2 grans tasques:

1. Cerca de *bugs*
2. Correcció de *bugs*

Que com bé diu el seu nom, primer es farà una cerca de *bugs*, per realitzar aquesta tasca, es demanarà a gent externa al projecte que faci proves per tal de trobar el màxim de *bugs* possibles, i un cop trobats, a mesura que es van trobant (per tant la cerca no és precedent) s'aniran corregint.

Documentació

Durant l'última fase es realitzarà la redacció de la memòria del projecte i la presentació final, això estarà dividit bàsicament en 2 tasques:

1. Memòria final
2. Presentació defensa projecte

I es realitzarà un cop trobats tots els *bugs* i de manera paral·lela a la correcció d'aquests.

7.2 Recursos

Recursos Humans

Com a recursos humans es disposarà únicament d'una desenvolupadora que serà l'encarregada de realitzar totes les tasques, addicionalment es disposarà d'ajuda externa com per exemple per la cerca de bugs de part de companys d'universitat.

Recursos Hardware

Com a recursos de hardware s'utilitzarà:

- **Portàtil personal:** Dell inspiron 15 5000 series.
- **Recursos d'Amazon Web Services (AWS):** Per tal de realitzar l'API s'utilitzaran diversos recursos que es presenten més endavant al capítol de disseny del sistema

Recursos Software

Per part del software s'utilitzaran diversos programes:

- **WebStorm:** Editor de text de l'empresa JetBrains que s'utilitzarà per escriure el codi del Front-End (escrit en JavaScript).
- **GoLand:** Editor de text de l'empresa JetBrains que s'utilitzarà per escriure el codi del Back-End (escrit en Go).
- **Postman:** Per testejar l'API.
- **Consola d'AWS:** Per utilitzar els diferents serveis d'Amazon que s'utilitzaran per crear l'API de l'aplicació
- **React.js:** Framework que s'utilitzarà per realitzar el Front-End.
- **Git:** Controlador de versions per pujar el codi a un repositori de GitHub.
- **LaTeX:** Editor de textos per escriure tant els documents de GEP com la memòria final.
- **Swagger:** Editor de text que s'utilitzarà per documentar els endpoints a desenvolupar.

7.3 Estimació de temps

L'estimació de temps es pot veure a la taula 7.1 on dins del desenvolupament tant de Back-End com de Front-End estarà subdividit el temps per Sprints on es desenvoluparan les diferents històries d'usuari que es defineixen a l'inici de cada Sprint.

	Estimació de temps
Tasques inicials	20 hores
Desenvolupament Back-End	150 hores
Desenvolupament Front-End	200 hores
Millores	30 hores
Documentació	20 hores
TOTAL	420 hores

Taula 7.1: Estimació inicial de temps

7.4 Diagrama de Gantt

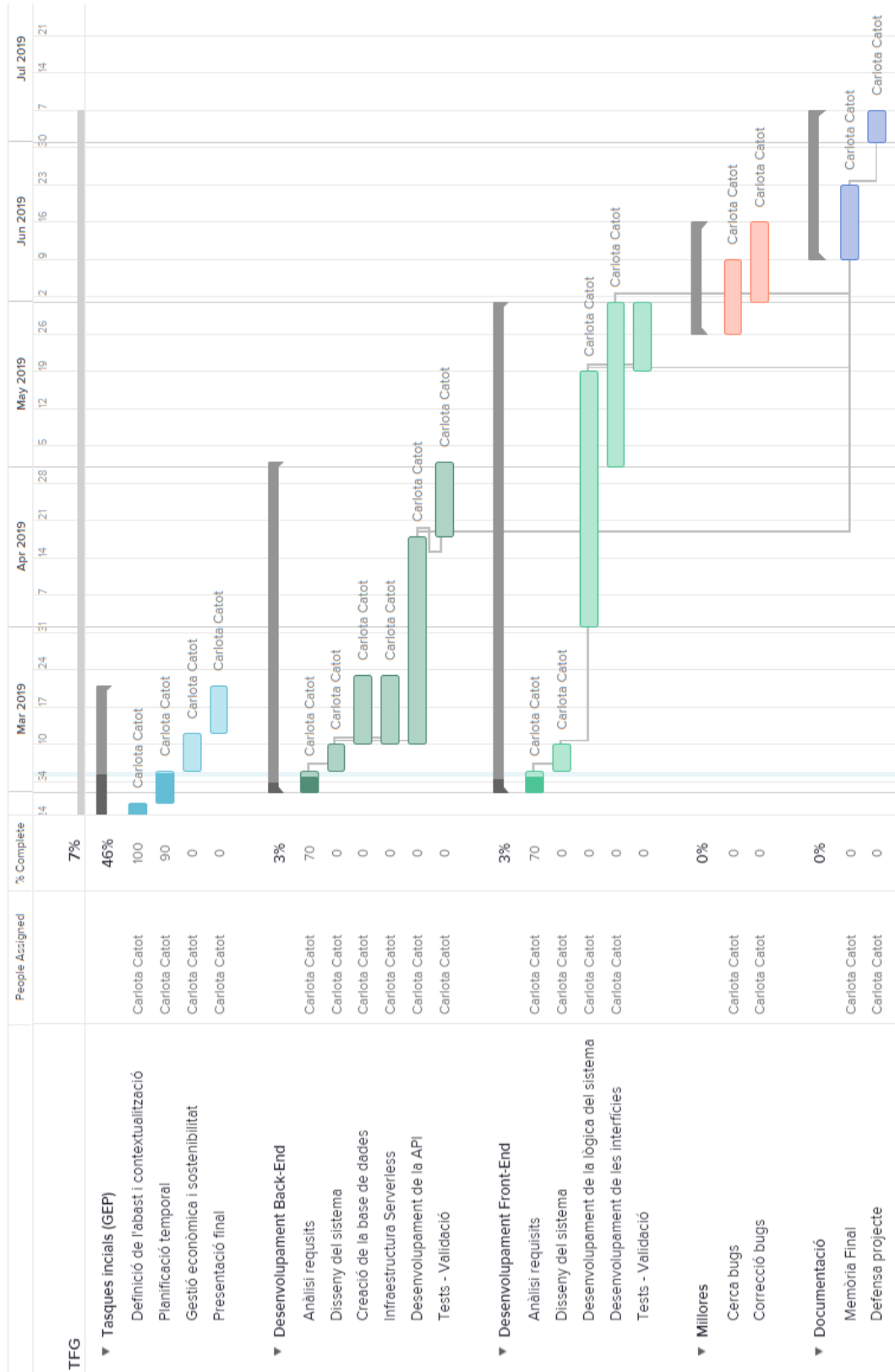


Figura 7.1: Diagrama de Gantt realitzat amb TeamGantt [11]

7.5 Pla d'acció

Com a pla d'acció per tal de complir els terminis indicats en els apartats anteriors, s'intentarà seguir les metodologies Agils per tal de poder complir els terminis, i en cas que s'acabi abans començar noves tasques o en el cas que s'acabi una tasca més tard realitzar les tasques següents el més ràpid possible. Per tal d'aconseguir un ritme òptim també es comptarà amb feedback per part de la directora del projecte per tal de solucionar més ràpidament els possibles problemes que es puguin trobar.

Els possibles endarreriments poden venir donats pel segon obstacle descrit a l'apartat 6.2.2, la complexitat, en cas de trobar la situació de veure que s'està perdent molt el temps a l'aprenentatge de Golang, es canviarà el llenguatge de programació de l'API a Node.js, ja que JavaScript és un llenguatge més conegut per la desenvolupadora del projecte. I en el cas de trobar un problema amb la complexitat d'una de les funcionalitats, s'intentarà reduir sense canviar l'objectiu final del projecte, però assegurant sempre que el projecte estarà acabat en el període establert.

Capítol 8

Planificació final

Finalment s'ha aconseguit assolir tots els períodes de temps tot i que els terminis s'han variat una mica, això es pot veure en el nou diagrama de gantt a la imatge 8.1, s'han conservat les mateixes tasques a realitzar però les tasques de Front-End i de Back-End s'han executat 100% en paral·lel, això ha vingut atès que es va començar dissenyant tot el front-end per tal de poder saber quins end-points eren necessaris per començar a desenvolupar l'API del sistema.

El temps que s'ha emprat per a la realització del projecte es pot veure a la taula 8.1 les hores no són 100% exactes, ja que aconseguir una exactitud en aquest aspecte és pràcticament impossible, però s'han calculat de la manera més exacta possible. Es pot veure que els temps marcats a la planificació inicial eren molt optimistes i això repercutirà a la gestió econòmica.

Per tal d'aconseguir les hores emprades a les tasques inicials i documentació s'ha calculat, de manera aproximada, el temps que s'ha dedicat a l'escriptura dels documents. Per les tasques de desenvolupament s'ha anat comptat, també de manera aproximada, el temps invertit en les tasques de cada sprint, com hi ha hagut feina no contemplada a les tasques, s'ha comptat temps invertit per Sprint (les hores de cada tasca es poden consultar al taiga de cada sprint). Aquestes tasques es troben explicades amb més detall al capítol de diari d'execució, on apareix un resum de les hores de desenvolupament i millores més desglossat, pel que fa a la documentació s'ha incrementat a 50 hores la realització d'aquest document i les múltiples versions que ha tingut.

	Hores emprades aproximadament
Tasques inicials	20 hores
Desenvolupament Back-End	175 hores
Desenvolupament Front-End	249 hores
Millores	25 hores
Documentació	50 hores
TOTAL	529 hores

Taula 8.1: Estimació inicial de temps

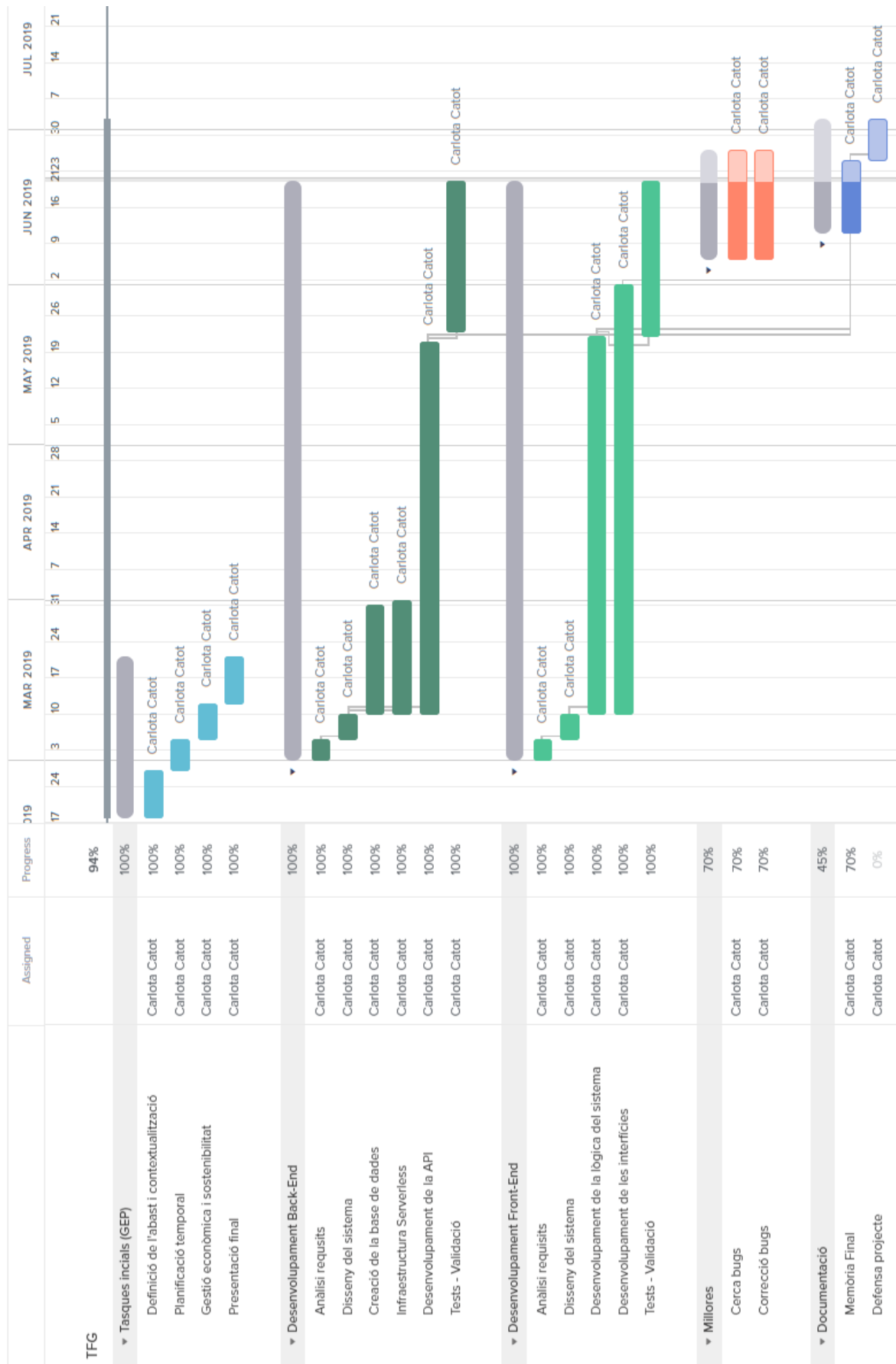


Figura 8.1: Diagrama de Gantt final realitzat amb TeamGantt [11]

Capítol 9

Gestió econòmica

A continuació, es realitzarà una estimació del pressupost que es necessita per tal de poder fer possible aquest projecte. Els costos es dividiran en tres grans categories: Costos directes (que implicaran els recursos esmentats a l'apartat 7.2), costos indirectes (que són recursos que són necessaris per realitzar el projecte però no es compta amb ells explícitament per això com pot ser la llum o l'internet), i per últim, costos imprevistos (que cobriran els possibles imprevistos que ens podem trobar durant el desenvolupament del treball). Com a punt final, trobarem un resum de tots els costos mencionats per tal d'indicar el cost total del projecte.

9.1 Costos Directes

Com a costos directes definim les despeses dels diferents recursos que s'utilitzen per a la realització del projecte, és a dir, el software, el hardware i els recursos humans. Com aquests recursos s'utilitzaran durant les diferents seccions que hem definit al Gantt durant la planificació, es calcularà l'amortització en el temps que duri aquella secció.

Per tal de calcular l'amortització en alguns dels recursos utilitzats, tindrem en compte dos factors: La vida útil del recurs i que el nostre projecte té una durada total de 5 mesos.

9.1.1 Tasques inicials (GEP)

Temps estimat per aquesta secció: 1 mes [20 hores] (del 20 de febrer al 20 de març) i el total del cost per aquesta secció el podem veure a la taula 9.1

Costos de Hardware			
Producte	Preu	Vida útil	Amortització
PC (Dell inspiron 15 5000 series)	758,00 €	5 anys (60 mesos)	12,63 €
Costos de Software			
Producte	Preu	Vida útil	Amortització
LaTeX	0,00 €	-	0,00 €
Costos Humans			
	Hores	Preu per hora	Sou
Desenvolupadora	20 hores	18,00 €	360,00 €
TOTAL			372,63 €

Taula 9.1: Cost bloc *tasques inicials (GEP)*

9.1.2 Desenvolupament Back-End

Temps estimat per aquesta secció: 2 mesos [150 hores] (de l'1 de març a l'1 de maig) i el total del cost per aquesta secció el podem veure a la taula 9.2

Costos de Hardware			
Producte	Preu	Vida útil	Amortització
PC (Dell inspiron 15 5000 series)	758,00 €	5 anys (60 mesos)	25,27 €
Amazon Web Services	0,00 €	-	0,00 €
Costos de Software			
Producte	Preu	Vida útil	Amortització
GoLand	0,00 €	-	0,00 €
Swagger	0,00 €	-	0,00 €
Consola AWS	0,00 €	-	0,00 €
Postman	0,00 €	-	0,00 €
Git	0,00 €	-	0,00 €
Costos Humans			
	Hores	Preu per hora	Sou
Desenvolupadora	150 hores	18,00 €	2.700,00 €
TOTAL			2.725,27 €

Taula 9.2: Cost bloc *desenvolupament Back-End*

9.1.3 Desenvolupament Front-End

Temps estimat per aquesta secció: 3 mesos [200 hores] (del 1 de març a l'31 de maig) i el total del cost per aquesta secció el podem veure a la taula 9.3

Costos de Hardware			
Producte	Preu	Vida útil	Amortització
PC (Dell inspiron 15 5000 series)	758,00 €	5 anys (60 mesos)	12,63 € *
Costos de Software			
Producte	Preu	Vida útil	Amortització
WebStorm	0,00 €	-	0,00 €
React.js	0,00 €	-	0,00 €
Git	0,00 €	-	0,00 €
Costos Humans			
	Hores	Preu per hora	Sou
Desenvolupadora	200 hores	18,00 €	3.600,00 €
TOTAL			3.612,63 €

Taula 9.3: Cost bloc *desenvolupament Front-End*

* Tot i que l'amortització en aquesta secció hauria de ser de 37,89 €, com durant els dos primers mesos, aquesta secció passa durant el mateix període de temps que el desenvolupament de Back-End, només es s'ha comptat el preu del mes extra.

9.1.4 Millores

Temps estimat per aquesta secció: 0,5 mesos [30 hores] (del 31 de maig al 15 de juny) i el total del cost per aquesta secció el podem veure a la taula 9.4

Costos de Hardware			
Producte	Preu	Vida útil	Amortització
PC (Dell inspiron 15 5000 series)	758,00 €	5 anys (60 mesos)	6,315 €
Costos de Software			
Producte	Preu	Vida útil	Amortització
GoLand	0,00 €	-	0,00 €
Swagger	0,00 €	-	0,00 €
Consola AWS	0,00 €	-	0,00 €
Postman	0,00 €	-	0,00 €
WebStorm	0,00 €	-	0,00 €
React.js	0,00 €	-	0,00 €
Git	0,00 €	-	0,00 €
Costos Humans			
	Hores	Preu per hora	Sou
Desenvolupadora	30 hores	18,00 €	540,00 €
TOTAL			546,32 €

Taula 9.4: Cost bloc *millores*

9.1.5 Documentació

Temps estimat per aquesta secció: 0,5 mesos [30 hores] (del 15 de juny al 30 de juny) i el total del cost per aquesta secció el podem veure a la taula 9.5

Costos de Hardware			
Producte	Preu	Vida útil	Amortització
PC (Dell inspiron 15 5000 series)	758,00 €	5 anys (60 mesos)	6,315 €
Costos de Software			
Producte	Preu	Vida útil	Amortització
LaTeX	0,00 €	-	0,00 €
Costos Humans			
	Hores	Preu per hora	Sou
Desenvolupadora	20 hores	18,00 €	360,00 €
TOTAL			366,31 €

Taula 9.5: Cost bloc *documentació*

9.1.6 Total costos directes

Després d'analitzar els costos de cada secció de tasques descrites a la planificació inicial, a la taula 9.6 tenim el total dels costos directes del projecte.

	Preu
Tasques inicials	372,63 €
Desenvolupament Back-End	2.725,27 €
Desenvolupament Front-End	3.612,63 €
Millores	546,32 €
Documentació	366,31 €
TOTAL	7.623,16 €

Taula 9.6: Taula dels costos directes

9.2 Costos Indirectes

Com a costos indirectes (taula 9.7) definim les despeses indirectes del projecte, aquestes són despeses que no s'han contractat directament per poder desenvolupar el projecte, però que sense elles no es podria realitzar el projecte.

Producte	Preu	Unitats	Preu estimat
Electricitat	0,14 €/kW	1500 kWh	210,00 €
Internet	30,00 €/mes	5 mesos	150,00 €
Espai	0,00 €	-	0,00 €
Altres	100,00 €	-	100,00 €
TOTAL			460,00 €

Taula 9.7: Taula dels costos indirectes

De la taula anterior cal destacar les dues últimes files, l'espai té el preu a 0, ja que el projecte es desenvoluparà a l'habitatge familiar del qual ja s'ha de pagar res i a l'apartat d'altres inclourem el transport, material d'oficina, l'ús d'altres ordinadors...

9.3 Costos Imprevistos

Segons els imprevistos definits anteriorment tant a la planificació com a la contextualització del projecte, podem necessitar un extra de temps per part de la desenvolupadora ja sigui perquè s'han calculat malament els temps i s'ha de fer un sprint final on s'han de dedicar més hores o perquè una de les tasques complicades hi han necessitat més hores de desenvolupament.

Comptant que el preu per hora definit per la feina de desenvolupadora del projecte és de 18 €/hora es pot comptar que si s'allarga unes 100 hores com al pitjor dels casos, podem guardar **1.800,00 €** per costos imprevistos.

9.4 Costos Totals

Finalment per trobar el cost total del projecte (taula 9.8) sumarem els costos definits anteriorment afegint una contingència del 5% per tal de cobrir les possibles despeses inesperades que poden tenir lloc durant el desenvolupament del projecte.

	Preu
Costos Directes	7.623,16 €
Costos Indirectes	460,00 €
Costos Imprevistos	1.800,00 €
Subtotal	9.883,16 €
Contingència (5%)	494,158 €
TOTAL	10.377,32 €

Taula 9.8: Taula resum dels costos del projecte

9.5 Control de pressupost

Per al control de pressupost caldrà controlar les hores de desenvolupament i per aquest motiu, en finalitzar cadascuna de les seccions definides s'escriurà el temps en hores real que s'ha dedicat a realitzar les diferents tasques i així poder calcular el preu real. D'aquesta manera, es calcularà la diferència entre les hores estimades i les hores reals i en multiplicar-ho pel cost estimat obtindrem la desviació de consum. També es calcularà la desviació del cost mitjançant la diferència entre el cost real i el cost estimat multiplicat pel nombre d'hores reals. Per sintetitzar podem definir les següents fórmules:

$$\text{Desviació de consum} = (\text{Hores estimades} - \text{Hores reals}) * \text{Cost estimat}$$

$$\text{Desviació de cost} = (\text{Cost estimat} - \text{Cost real}) * \text{Hores reals}$$

Mitjançant aquest procés podrem observar d'una manera fàcil les desviacions que hi ha hagut i així poder determinar si es degut a un augment (o decrement en el millor dels casos) del nombre d'hores o dels costos.

Capítol 10

Cost real del projecte

Després de tot el desenvolupament i tenint en compte les hores emprades presentades a la planificació final es presenta el cost final del projecte on ja no caldrà afegir ni gastos imprevistos ni contingències perquè es considera un projecte tancat.

10.1 Costos directes reals

10.1.1 Tasques inicials (GEP)

Costos de Hardware			
Producte	Preu	Vida útil	Amortització
PC (Dell inspiron 15 5000 series)	758,00 €	5 anys (60 mesos)	12,63 €
Costos de Software			
Producte	Preu	Vida útil	Amortització
LaTeX	0,00 €	-	0,00 €
Costos Humans			
	Hores	Preu per hora	Sou
Desenvolupadora	20 hores	18,00 €	360,00 €
TOTAL			372,63 €

Taula 10.1: Cost reals bloc *tasques inicials (GEP)*

10.1.2 Desenvolupament Back-End

Costos de Hardware			
Producte	Preu	Vida útil	Amortització
PC (Dell inspiron 15 5000 series)	758,00 €	5 anys (60 mesos)	25,27 €
Amazon Web Services	0,00 €	-	0,00 €
Costos de Software			
Producte	Preu	Vida útil	Amortització
GoLand	0,00 €	-	0,00 €
Swagger	0,00 €	-	0,00 €
Consola AWS	0,00 €	-	0,00 €
Postman	0,00 €	-	0,00 €
Git	0,00 €	-	0,00 €
Costos Humans			
	Hores	Preu per hora	Sou
Desenvolupadora	175 hores	18,00 €	3.150,00 €
TOTAL			3.175,27 €

Taula 10.2: Cost real bloc *desenvolupament Back-End*

10.1.3 Desenvolupament Front-End

Costos de Hardware			
Producte	Preu	Vida útil	Amortització
PC (Dell inspiron 15 5000 series)	758,00 €	5 anys (60 mesos)	12,63 €*
Costos de Software			
Producte	Preu	Vida útil	Amortització
WebStorm	0,00 €	-	0,00 €
React.js	0,00 €	-	0,00 €
Git	0,00 €	-	0,00 €
Costos Humans			
	Hores	Preu per hora	Sou
Desenvolupadora	249 hores	18,00 €	4.482,00 €
TOTAL			4.494,63 €

Taula 10.3: Cost real bloc *desenvolupament Front-End*

10.1.4 Milllores

Costos de Hardware			
Producte	Preu	Vida útil	Amortització
PC (Dell inspiron 15 5000 series)	758,00 €	5 anys (60 mesos)	6,315 €
Costos de Software			
Producte	Preu	Vida útil	Amortització
GoLand	0,00 €	-	0,00 €
Swagger	0,00 €	-	0,00 €
Consola AWS	0,00 €	-	0,00 €
Postman	0,00 €	-	0,00 €
WebStorm	0,00 €	-	0,00 €
React.js	0,00 €	-	0,00 €
Git	0,00 €	-	0,00 €
Costos Humans			
	Hores	Preu per hora	Sou
Desenvolupadora	25 hores	18,00 €	450,00 €
TOTAL			456,32 €

Taula 10.4: Cost real bloc *milllores*

10.1.5 Documentació

Costos de Hardware			
Producte	Preu	Vida útil	Amortització
PC (Dell inspiron 15 5000 series)	758,00 €	5 anys (60 mesos)	6,315 €
Costos de Software			
Producte	Preu	Vida útil	Amortització
LaTeX	0,00 €	-	0,00 €
Costos Humans			
	Hores	Preu per hora	Sou
Desenvolupadora	50 hores	18,00 €	900,00 €
TOTAL			906,31 €

Taula 10.5: Cost real bloc *documentació*

10.2 Total costos directes

	Preu
Tasques inicials	372,63 €
Desenvolupament Back-End	3.175,27 €
Desenvolupament Front-End	4.494,63 €
Millores	456,32 €
Documentació	906,31 €
TOTAL	9.405,16 €

Taula 10.6: Taula dels costos directes

10.3 Costos Totals

Finalment per trobar el cost real del projecte (taula 10.7) i sabent que els costos indirectes no han variat, i els costos directes son els que s'han descrit anteriorment.

	Preu
Costos Directes	9.405,16 €
Costos Indirectes	460,00 €
TOTAL	9.865,16 €

Taula 10.7: Taula resum dels costos del projecte

$$\text{Desviació de consum} = (420 \text{ hores} - 529 \text{ hores}) * 10.377,32\text{€} = -1.131.127,88$$

$$\text{Desviació de cost} = (10.377,32\text{€} - 9.865,16\text{€}) * 529 = 270.932,64$$

Finalment podem veure que el projecte ha sortit més rendible del que s'esperava tot i tenir una desviació en consum molt alta i una desviació en cost no tant.

Capítol 11

Sostenibilitat i impacte social

	Ambiental	Social	Económic
PPP	8/10	9/10	9/10
Vida útil	8/10	9/10	8/10
Riscos	9/10	9/10	7/10

Taula 11.1: Matriu de sostenibilitat

11.1 Dimensió ambiental

A la dimensió ambiental del projecte podem destacar que tot el sistema està desenvolupat utilitzant l'energia d'un únic ordinador, que pertany a la desenvolupadora d'aquest, ja que els servidors que s'utilitzen es troben al DataCenter d'Amazon Web Services el que fa que no es consumeixi un excés de kWh, ja que en contractar el sistema d'AWS es comparteix servidor amb altres aplicacions, ja que l'optimització d'aquests recursos es fa des del mateix Cloud.

Per tant, per la dimensió ambiental podem definir que el nostre projecte utilitzant el Cloud no tindrà gaire consum elèctric per tant el valor del projecte posat en producció és positiu. Tampoc tindrà petjada ecològica, ja que la contaminació que pugui generar el DataCenter no serà causat per aquesta aplicació, ja que el DataCenter ja existia anteriorment, per tant pel que fa a la vida útil del projecte en el sentit ambiental també es dona un valor positiu. Per últim, els riscos ambientals d'aquesta aplicació, com hem comentat abans, no depèn de l'aplicació sinó del DataCenter o de quants desenvolupadors estiguin consumint energia, però per al moment es considera molt positiu, ja que no causa cap problema al medi ambient.

11.2 Dimensió social

Pel que fa a la dimensió social per l'apartat del projecte posat en producció, podem dividir-la en dues parts, com afecta el projecte als desenvolupadors de l'aplicació i com afecta aquesta aplicació als seus usuaris.

En primer lloc, als desenvolupadors del projecte els aportarà coneixements nous tant de desenvolupament com de gestió de projectes, ja que és un projecte gran que ha de ser desenvolupat en poc temps.

Pels usuaris de l'aplicació (els clients) l'ús de l'aplicació els ajudarà a tenir un control més acurat de les seves despeses per tal de poder estalviar millor, aquest aspecte social es considera molt positiu, ja que ajuda a solucionar un gran problema que pot ser la manca de diners a final de mes a causa d'un excés de consumisme.

Un cop analitzat l'impacte social en els implicats en el projecte, podem concloure que el projecte es pot considerar útil i que aporta positivament a la societat.

Pel que fa a la vida útil del projecte sobre Impacte social, es considera positiu, ja que com s'ha comentat anteriorment pels usuaris de l'aplicació, l'ús de l'aplicació els ajudarà a tenir un control més acurat de les seves despeses per tal de poder estalviar millor, i com s'ha comentat també prèviament als apartats anteriors, això pot arribar a fer que millori l'economia del país en general, per tant es dona un valor positiu.

Per últim els riscos socials que pot tenir l'aplicació són pocs, ja que és bastant personal, les dades es troben encriptades i per al moment no hi ha connexions externes. El que podríem destacar és que poden sorgir problemes en el cas dels grups, però això s'escapa de les competències del projecte per tant es dona un valor positiu en aquest camp.

11.3 Dimensió Econòmica

Pel que fa a la dimensió econòmica, les despeses previstes i el càlcul total del desenvolupament del projecte fins al moment es troben explicats als apartats anteriors. Podem destacar que s'ha decidit posar uns valors baixos, ja que com es pot veure amb la gestió econòmica el projecte no sortirà car, per tant per l'apartat de projecte posat en producció es considera que el valor de la factura és prou bo i es dona un valor baix.

Pels camps de vida útil i riscos s'ha de comptar amb el detall que AWS fa els cobraments per números de petició per tant els valors de la factura es dispararien en cas d'utilitzar de manera molt recurrent i amb molts usuaris l'aplicació. També s'ha de tenir en compte que en cas que és dones aquest cas, l'aplicació donaria benefici que en principi compensarien totes les despeses, per tant, com es considera que és una aplicació que podria donar benefici, té una bona viabilitat i es dona un valor positiu pel camp de vida útil. Seguint amb la mateixa línia els riscos econòmics són que l'aplicació no funcioni, valor que va molt lligat al de la vida útil, per tant de manera optimista també se li ha donat un valor positiu.

Capítol 12

Requisits

Per tal de definir els requisits de l'aplicació s'han agafat les històries d'usuari especificades al llarg del desenvolupament i s'ha descrit els requisits funcionals i els criteris d'acceptació per cadascuna d'aquestes històries d'usuari, a excepció de les històries del primer (ja que en ser únicament de configuració del sistema no tenen uns requisits marcats) i l'últim Sprint (ja que són les millors tant en la interfície com en el funcionament del sistema del qual ja s'han especificat els requisits). Els requisits no funcionals són genèrics per totes les funcionalitats del sistema:

12.1 Requisits no funcionals

En aquest apartat es destaquen els 3 requisits no funcionals bàsics que ha de complir el sistema.

Aparença

- **Tipus de requisit segons la classificació Volere:** 10a. Appearance
- **Descripció:** Disseny atractiu perquè els usuaris es sentin còmodes utilitzant l'aplicació.
- **Justificació del requisit:** Com que l'aplicació està destinada a qualsevol mena de persones, és necessari que el sistema sigui atractiu i usable per poder atraure l'atenció d'aquest i la vulguin tornar a utilitzar.
- **Condicció de satisfacció:** Durant les proves de testing el feedback sobre el disseny és positiu.

Usabilitat

- **Tipus de requisit segons la classificació Volere:** 11a. Ease of Use
- **Descripció:** L'aplicació ha de ser de fàcil ús per tal que els usuaris la vulguin seguir utilitzant.
- **Justificació del requisit:** Perquè l'aplicació sigui útil és necessari que els usuaris la vulguin fer servir perquè vagin actualitzant periòdicament les seves dades
- **Condicció de satisfacció:** En realitzar el testing el 90% dels usuaris que han utilitzat l'app (no tots han reportat errors), asseguren que la tornarien a utilitzar.

Seguretat

- **Tipus de requisit segons la classificació Volere:** 15c. Privacy requirements
- **Descripció:** L'aplicació assegurarà la privadesa de les dades dels usuaris.
- **Justificació del requisit:** S'han de protegir les dades dels usuaris de manera que només ells tinguin accés directe a aquests, les contrasenyes han d'estar encriptats.

- **Condició de satisfacció:** Cada usuari pot accedir només a les seves dades personals, quant als grups pot veure la informació bàsica de cada contacte sense afectar a la privacitat.

12.2 Requisits funcionals

En aquest subapartat es presenten els requisits funcionals de l'aplicació, organitzats per les diferents històries d'usuari, entre parèntesis s'indica l'Sprint en el qual s'ha desenvolupat cada història.

Login i Logout (Sprint 2) - un usuari accedeix al sistema o tancar la sessió activa

1. La connexió amb Amazon Cognito es realitza correctament amb per tal de verificar les credencials de l'usuari, en el cas de login, i per tancar la sessió, en el cas de logout.
2. S'accedeix al sistema mitjançant el nom d'usuari i una contrasenya correctament verificats.
3. Al fer correctament el login es guarden a les cookies les dades necessàries per a poder utilitzar l'aplicació posteriorment (Token, username), aquestes cookies s'esborraran al fer logout.

Registre (Sprint 2) - un usuari vol registrar-se al sistema

1. La connexió amb Amazon Cognito es realitza correctament amb per tal de crear l'usuari i verificar les seves dades.
2. L'usuari es registra mitjançant un correu electrònic, un nom d'usuari i una contrasenya introduïda dues vegades per tal d'assegurar que és correcta.

Gràfics (Sprint 3) - un usuari vol veure de manera gràfica les seves despeses

1. S'han de mostrar les gràfiques per les estadístiques de l'any.
2. Les dades s'han de mostrar per categories i separades entre despeses i benefici.

Gestió de despeses (Sprint 3) - un usuari vol gestionar les seves despeses

1. La connexió de l'API gateway amb la base de dades de DynamoDB funciona de manera correcta per tal de tenir totes les dades actualitzades.
2. La connexió del Front-End amb el Back-End (API gateway) mostra les dades correctament.
3. Una transacció ha de tenir els camps de: Data, títol, descripció, categoria (especificades a continuació), quantitat i tipus (benefici o despesa).
 - Per una transacció considerada benefici, les categories disponibles seran:
 - (a) Altres
 - (b) Alimentació i llar
 - (c) Assegurances
 - (d) Bellesa i higiene
 - (e) Despeses habitatge
 - (f) Oci i cultura
 - (g) Regals
 - (h) Sanitat
 - (i) Transport
 - (j) Vehicle
 - Per una transacció considerada despesa, les categories disponibles seran:
 - (a) Efectiu

- (b) Nòmina
- (c) Transferència

4. L'usuari ha de poder crear una transacció amb els camps anomenats anteriorment.
5. L'usuari ha de poder filtrar les transaccions per data, títol, descripció, categoria o tipus.
6. L'usuari ha de poder ordenar les transaccions per qualsevol camp definit anteriorment.
7. L'usuari ha de poder eliminar una transacció.
8. L'usuari ha de poder editar els camps de títol, descripció, quantitat i/o categoria d'una transacció.
9. L'usuari ha de poder veure les transaccions per anys.

Perfil d'usuari (Sprint 4) - un usuari vol gestionar el seu perfil

1. Les dades carregades a la pantalla s'extreuen de cognito tot i que també hi ha una base de dades on es guarden les dades bàsiques de l'usuari.
2. Un usuari ha de tenir com a camps obligatoris nom d'usuari i e-mail, i pot tenir de manera opcional els camps de nom i cognom.
3. L'usuari ha de poder modificar els camps de nom i cognom sempre que vulgui. Les dades s'han de modificar tant a cognito com a la base de dades.
4. L'usuari no ha de poder modificar els camps de nom d'usuari i e-mail.
5. L'usuari ha de poder modificar la seva contrasenya sempre que vulgui introduint la contrasenya actual.

Pantalla principal (Sprint 4) - un usuari vol una pantalla amb les dades més significatives

1. L'usuari ha de poder veure el seu saldo actual que s'actualitzarà cada cop que es fa un canvi a la pantalla d'activitat.
2. L'usuari ha de poder veure les estadístiques dels grups (el total de què deu i el que se li deu)
3. L'usuari ha de poder veure les estadístiques de les despeses fetes durant l'any actual i les gràfiques on mostren les despeses per categories.

Gestió de contactes (Sprint 5) - un usuari vol gestionar els seus contactes

1. L'usuari ha de veure tots els contactes que ha afegit ell o els usuaris que el tenen com a contacte, no hi ha diferenciació.
2. L'usuari pot afegir un contacte en qualsevol moment, sempre que aquest estigui registrat al sistema.
 - (a) A l'hora d'afegir un contacte l'usuari ha de poder veure un llistat dels usuaris creats per tal d'afegir un usuari vàlid.
3. L'usuari pot esborrar un contacte en qualsevol moment.
4. L'usuari ha de poder veure per cada usuari els comptes pendents que té amb ell.

Gestió de grups (Sprint 5) - un usuari vol gestionar els meus grups i els moviments de cadascun

1. Un grup ha de tenir els camps de títol, descripció i participants (l'usuari que crea el grup automàticament sempre serà membre del grup).
2. L'usuari ha de poder crear un grup introduint les dades necessàries.
3. L'usuari ha de poder editar la informació del grup.
4. L'usuari ha de poder esborrar un grup.
5. L'usuari ha de poder consultar els deutes que té amb tots els participants d'aquest i els que tenen tots els participants entre ells.
6. L'usuari ha de poder afegir un moviment al grup. Un moviment ha de tenir els camps de data, títol, descripció, pagador, participants i quantitat.
7. L'usuari ha de poder filtrar els moviments per qualsevol dels camps i també ordenar-los.
8. L'usuari ha de poder editar o esborrar un moviment.
9. L'usuari ha de poder liquidar els deutes pendents que té amb els altres usuaris.

Exportar (Sprint 6) - un usuari vol exportar la taula de despeses a Excel

1. L'usuari ha de poder exportar en Excel la taula de totes les despeses, i també la taula de moviments d'un grup.

Multiidioma (Sprint 6) - un usuari vol poder consultar la web en el seu idioma

1. L'aplicació d'un inici s'ha de presentar íntegrament en anglès.
2. L'aplicació ha d'oferir a l'usuari l'opció d'escollir entre català, castellà o anglès.

Versió mòbil (Sprint 6) - un usuari vol poder consultar l'aplicació des de qualsevol dispositiu

1. L'aplicació ha d'oferir un sistema mòbil que ofereixi les funcionalitats bàsiques, que són les de fer consultes i afegir informació. Per tal de consultar gràfiques, exportar en Excel o editar/esborrar, serà necessari utilitzar la versió web.

12.3 Criteris d'acceptació

Els criteris d'acceptació són els requisits que indiquen de quina manera ha de comportar-se l'aplicació per cada funcionalitat d'aquesta. Aquests criteris han de descriure sempre un context, un esdeveniment i una resposta o conseqüència esperada pel sistema, per tal de definir-los s'ha utilitzat l'estructura de **Donat-Quan-Aleshores** [12]. A continuació es presenten els criteris d'acceptació per cada història d'usuari del sistema, i com a l'apartat anterior, entre parèntesis s'indica l'Sprint en el qual s'ha desenvolupat cada història.

Login i Logout (Sprint 2) - un usuari accedeix al sistema o tancar la sessió activa

Com a usuari **vull** accedir o tancar el sistema **per** a poder començar a utilitzar l'aplicació o acabar d'utilitzar-la.

1. **Donada** la pantalla login de l'aplicació, **quan** l'usuari clica sobre els camps editables **aleshores** apareixerà el cursor de manera que l'usuari podrà començar a teclejar de manera immediata.

2. **Donada** la pantalla login de l'aplicació, **quan** els camps nom d'usuari i contrasenya han estat omplerts i l'usuari prem el botó *Entrar aleshores* es comproven les credencials al servidor on es troben les dades de tots els usuaris, en cas que siguin correctes s'entra a la pàgina principal, si no es mostra un error i permet a l'usuari tornar a introduir les dades.
3. **Donada** la pantalla login de l'aplicació, **quan** l'usuari clica sobre *Registrar-me ara aleshores* es redirigeix cap a la pantalla de registre.
4. **Donada** la pantalla principal de l'aplicació, **quan** l'usuari clica sobre *Logout* a la barra de menú situada a la part superior de la pantalla, **aleshores** es tanca la sessió iniciada i l'usuari torna a la pantalla de login.

Registre (Sprint 2) - un usuari vol registrar-se al sistema

Com a usuari **vull** registrar-me al sistema **per** tal de poder utilitzar l'aplicació.

1. **Donada** la pantalla registre de l'aplicació, **quan** l'usuari clica sobre els camps editables **aleshores** apareixerà el cursor de manera que l'usuari podrà començar a teclejar de manera immediata.
2. **Donada** la pantalla registre de l'aplicació, **quan** els camps: nom d'usuari, e-mail i contrasenya han estat omplerts (els dos camps de les contrasenyes coincideixen) i l'usuari prem el botó *Registrar aleshores* es comproven les credencials al servidor on es troben les dades de tots els usuaris, en cas que l'usuari no pertanyi al sistema, es crearà un compte i s'enviarà un correu electronic a l'usuari per tal de verificar el compte.
3. **Donada** la pantalla de verificació de compte l'aplicació, **quan** l'usuari clica sobre els camps editables **aleshores** apareixerà el cursor de manera que l'usuari podrà començar a teclejar de manera immediata.
4. **Donada** la pantalla de verificació de compte de l'aplicació **quan** el camp del codi de validació ha estat omplert i l'usuari prem el botó *Validar*, **aleshores** es comprova si el codi és correcte, en cas afirmatiu l'usuari ha estat correctament creat i es pot començar a utilitzar l'aplicació.

Gràfics (Sprint 3) - un usuari vol veure de manera gràfica les seves despeses

Com a usuari **vull** poder tenir gràfiques de les meves despeses **per** tal de veure d'una manera més gràfica en què m'he gastat els diners.

1. **Donada** la pantalla d'activitat de l'aplicació, apartat de les transaccions per any **quan** l'usuari entra a la pantalla **aleshores** ha de poder veure les gràfiques on es mostra per categories el que s'ha gastat en cadascuna.

Gestió de despeses (Sprint 3) - un usuari vol gestionar les seves despeses

Com a usuari **vull** poder consultar, crear, editar o esborrar les meves despeses **per** tal de poder gestionar-les.

1. **Donada** la pantalla d'activitat de l'aplicació, **quan** l'usuari entra a la pantalla **aleshores** apareix una llista amb totes les despeses realitzades.
2. **Donada** la pantalla d'activitat de l'aplicació, **quan** selecciona l'opció de transaccions per any **aleshores** apareix una llista amb totes les despeses realitzades durant l'any seleccionat en un dropdown.
3. **Donada** la pantalla d'activitat de l'aplicació (on estan totes les transaccions) **quan** l'usuari selecciona el botó d'*editar* **aleshores** els camps d'aquella fila es tornen editables per tal de poder canviar-los.

- (a) **Donada** la fila editable **quan** els camps editables (títol, descripció, categoria i quantitat) estan omplerts i l'usuari clica sobre el botó de *guardar aleshores* es fan les gestions pertinents al Back-End per tal de modificar la transacció.
- 4. **Donada** la pantalla d'activitat de l'aplicació (on estan totes les transaccions) **quan** l'usuari selecciona el botó d'*esborrar aleshores* es fan les gestions pertinents al Back-End per tal d'esborrar la transacció.
- 5. **Donada** la pantalla d'activitat de l'aplicació (on estan totes les transaccions) **quan** l'usuari selecciona el botó d'*afegir transacció* (despesa o benefici) **aleshores** apareix un modal on l'usuari podrà afegir la informació per la nova transacció.
 - (a) **Donat** el modal per afegir una transacció **quan** l'usuari clica sobre els camps editables **aleshores** apareixerà el cursor de manera que l'usuari podrà començar a teclejar de manera immediata.
 - (b) **Donat** el modal per afegir una transacció **quan** tots els camps han estat omplerts i l'usuari prem sobre el botó de *crear aleshores* es fan les gestions pertinents al Back-End per tal de crear una nova transacció.

Perfil d'usuari (Sprint 4) - un usuari vol gestionar el seu perfil

Com a usuari **vull** poder consultar i modificar les dades del meu perfil **per** tal de tenir-les actualitzades pel que puguin veure els meus contactes.

- 1. **Donada** la pantalla de perfil de l'usuari, **quan** l'usuari entra a la pantalla **aleshores** apareixen les dades del seu perfil algunes modificables.
- 2. **Donada** la pantalla de perfil de l'usuari, **quan** l'usuari clica sobre els camps editables **aleshores** apareixerà el cursor de manera que l'usuari podrà començar a teclejar de manera immediata.
- 3. **Donada** la pantalla de perfil de l'usuari, **quan** tots els camps han estat omplerts i l'usuari prem sobre el botó de *guardar canvis aleshores* es fan les gestions pertinents al Back-End (tant a la base de dades com a cognito) per tal d'editar el perfil de l'usuari.
- 4. **Donada** la pantalla de perfil de l'usuari, **quan** l'usuari prem sobre el botó de *canviar contrasenya*, **aleshores** apareix un modal on l'usuari podrà modificar la seva contrasenya.
- 5. **Donat** el modal de canvi de contrasenya **quan** l'usuari clica sobre els camps editables **aleshores** apareixerà el cursor de manera que l'usuari podrà començar a teclejar de manera immediata.
- 6. **Donat** el modal de canvi de contrasenya **quan** tots els camps han estat omplerts (els dos camps de nova contrasenya coincideixen) i l'usuari prem sobre el botó de *canviar aleshores* es fan les gestions pertinents al Back-End per tal de canviar la contrasenya.

Pantalla principal (Sprint 4) - un usuari vol una pantalla amb les dades més significatives

Com a usuari **vull** tenir una pantalla principal **per** tal de tenir un resum amb les dades més significatives.

- 1. **Donada** la pantalla principal de l'aplicació, **quan** l'usuari entra a la pantalla **aleshores** apareixen les dades més significatives com són el saldo actual, les estadístiques dels grups i les estadístiques de l'any actual.

Gestió de contactes (Sprint 5) - un usuari vol gestionar els seus contactes

Com a usuari **vull** poder afegir o esborrar usuaris de la meua llista de contacte **per** tal de poder compartir despeses amb ells.

1. **Donada** la pantalla de contactes, **quan** l'usuari entra a la pantalla **aleshores** apareix una llista amb tots els contactes que té, ja sigui perquè l'ha afegit l'usuari o l'han afegit a ell.
2. **Donada** la pantalla de contactes, **quan** l'usuari selecciona el botó d'*afegir contacte* **aleshores** apareix un modal on l'usuari podrà afegir un contacte introduint un nom d'usuari vàlid.
 - (a) **Donat** el modal per afegir un contacte **quan** l'usuari clica sobre els camps editables **aleshores** apareixerà el cursor de manera que l'usuari podrà començar a teclejar de manera immediata. També apareixerà un llistat amb els usuaris que pertanyen al sistema.
 - (b) **Donat** el modal per afegir un contacte **quan** tots els camps han estat omplerts i l'usuari prem sobre el botó d'*afegir* **aleshores** es fan les gestions pertinents al Back-End per tal d'afegir al contacte.
3. **Donada** la pantalla de contactes **quan** l'usuari selecciona el botó d'*esborrar* **aleshores** es fan les gestions pertinents al Back-End per tal d'esborrar el contacte en qüestió.

Gestió de grups (Sprint 5) - un usuari vol gestionar els meus grups i els moviments de cadascun

Com a usuari **vull** poder gestionar grups **per** tal de dividir les despeses realitzades entre els participants d'aquests.

1. **Donada** la pantalla de grups, **quan** l'usuari entra a la pantalla **aleshores** apareix una llista amb tots els grups als quals pertany i els deutes que té amb cada participant.
2. **Donada** la pantalla de grups, **quan** l'usuari selecciona el botó de *crear grup* **aleshores** apareix modal on l'usuari podrà crear un grup introduint les dades corresponents.
 - (a) **Donat** el modal per crear un grup **quan** l'usuari clica sobre els camps editables **aleshores** apareixerà el cursor de manera que l'usuari podrà començar a teclejar de manera immediata. També apareixerà un llistat amb els usuaris que pertanyen al sistema.
 - (b) **Donat** el modal per crear un grup **quan** l'usuari clica sobre el botó d'*afegir participant* **aleshores** s'afegirà automàticament un camp per afegir un nou usuari.
 - (c) **Donat** el modal per crear un grup **quan** tots els camps han estat omplerts i l'usuari prem sobre el botó *crear* **aleshores** es fan les gestions pertinents al Back-End per tal de crear el grup.
3. **Donada** la llista de grups dins la pantalla general de grups, **quan** l'usuari selecciona el botó d'*esborrar* **aleshores** es fan les gestions pertinents al Back-End per tal d'esborrar el grup en qüestió.
4. **Donada** la llista de grups dins la pantalla general de grups, **quan** l'usuari selecciona el botó de *veure els moviments* **aleshores** s'obre una nova pestanya del navegador amb la pantalla d'informació completa del grup.
5. **Donada** la pantalla d'informació completa del grup, **quan** l'usuari accedeix a la pantalla **aleshores** apareix una taula amb tots els moviments que s'han realitzat en el grup.
6. **Donada** la pantalla d'informació completa del grup, **quan** l'usuari selecciona el botó d'*afegir moviment* **aleshores** apareix modal on l'usuari podrà afegir un moviment al grup introduint les dades corresponents.
 - (a) **Donat** el modal per afegir un moviment **quan** l'usuari clica sobre els camps editables **aleshores** apareixerà el cursor de manera que l'usuari podrà començar a teclejar de manera immediata. També apareixerà un llistat amb els usuaris que pertanyen al sistema.

- (b) **Donat** el modal per afegir un moviment **quan** l'usuari clica sobre el botó d'*afegir participant* **aleshores** s'afegirà automàticament un camp per afegir un nou usuari.
 - (c) **Donat** el modal per afegir un moviment **quan** tots els camps han estat omplerts i l'usuari prem sobre el botó *crear* **aleshores** es fan les gestions pertinents al Back-End per tal d'afegir el moviment.
7. **Donada** la pantalla d'informació completa del grup, **quan** l'usuari selecciona el botó d'*editar* **aleshores** els camps d'aquella fila es tornen editables per tal de poder canviar-los.
 - (a) **Donada** la fila editable **quan** els camps editables (títol, descripció, pagador i quantitat) estan omplerts i l'usuari clica sobre el botó de *guardar* **aleshores** es fan les gestions pertinents al Back-End per tal de modificar el moviment.
 8. **Donada** la pantalla d'informació completa del grup, **quan** l'usuari selecciona el botó d'*esborrar* **aleshores** es fan les gestions pertinents al Back-End per tal d'esborrar el moviment.
 9. **Donada** la pantalla d'informació completa del grup, **quan** l'usuari selecciona l'apartat de participants **aleshores** apareix un llistat amb els participants i els deutes que té cadascú amb la resta de participants del grup.
 - (a) **Donada** la fila de l'usuari actual **quan** prem el botó de *pagar* dins dels usuaris als quals deu diners **aleshores** es liquida el deute posant-lo a 0.
 10. **Donada** la pantalla d'informació completa del grup, **quan** l'usuari selecciona l'apartat d'informació **aleshores** apareixen camps per tal de poder editar la informació del grup de la mateixa manera que s'ha creat.

Exportar (Sprint 6) - un usuari vol exportar la taula de despeses a Excel

Com a usuari **vull** poder exportar en Excel la taula de despeses **per** tal de poder guardar la informació en local.

1. **Donada** la pantalla d'activitat de l'aplicació, **quan** l'usuari prem el botó *exportar a Excel* **aleshores** es descarrega la taula de totes les despeses en Excel.

Multiidioma (Sprint 6) - un usuari vol poder consultar la web en el seu idioma

Com a usuari **vull** poder consultar la web en el meu idioma **per** de poder entendre integralment el funcionament d'aquest.

1. **Donada** qualsevol pantalla de l'aplicació **quan** es selecciona un nou idioma de la llista d'idiomes disponibles **aleshores** automàticament es canvia l'idioma de tot el sistema.

Versió mòbil (Sprint 6) - un usuari vol poder consultar l'aplicació des de qualsevol dispositiu

Com a usuari **vull** que l'aplicació tingui versió mòbil **per** tal de poder consultar-la des de qualsevol dispositiu.

1. **Donada** qualsevol pantalla de l'aplicació **quan** s'està veient des de un dispositiu mòbil **aleshores** es veu una versió reduïda del sistema per tal de poder fer les operacions bàsiques.

Capítol 13

Disseny del sistema

Per tal d'explicar de quina manera s'ha dissenyat el sistema en aquest projecte podem basar-nos en l'esquema de la imatge 13.1, la idea d'utilitzar tecnologies d'AWS va sorgir fent una cerca de com fer una aplicació serverless tutorial1, tot i que no s'ha seguit 100% la informació explicada, s'han utilitzat les nocions bàsiques de disseny del sistema. Per començar podem veure que hi ha dues seccions clarament diferenciades, una pel Front-End i l'altra pel Back-End, per tant s'explicarà el disseny de les dues seccions per separat.

Pel que fa a la part del Front-End l'arquitectura cloud utilitzada és molt simple, s'ha creat una WebApp utilitzant React.js (amb components d'AntDesign), un framework de JavaScript, i el codi d'aquest s'ha anat pujant a GitHub. En primer lloc es va voler desplegar la webApp al servei S3 d'Amazon, es va cercar i trobar que es podia desplegar a través de travis [23], d'aquesta manera Travis s'encarrega de connectar el codi que es troba repositori de GitHub al sistema d'Amazon, així que cada cop que es fa un push a aquest repositori, es fa automàticament un desplegament del sistema. Posteriorment per tal d'utilitzar un sistema de CDN (Content Delivery Network) es va utilitzar CloudFront per crear les distribucions de l'aplicació i aquest serà el que ens doni l'URL per poder accedir a l'aplicació.

Pel que fa al Back-End, s'ha decidit fer una aplicació serverless i per això s'ha desenvolupat una API, com es pot veure a l'esquema, s'ha utilitzat l'API gateway per tal de crear tots els endpoint, el codi de cadascun es troba a una funció lambda i les dades estan emmagatzemades dins del sistema de bases de dades de dynamoDB (més endavant s'explica en detall les estructures que es troben a dynamoDB). Pel que fa als usuaris s'ha utilitzat el sistema de Cognito, que també ens serveix per poder autenticar a l'usuari a l'hora de fer peticions a l'API Gateway.

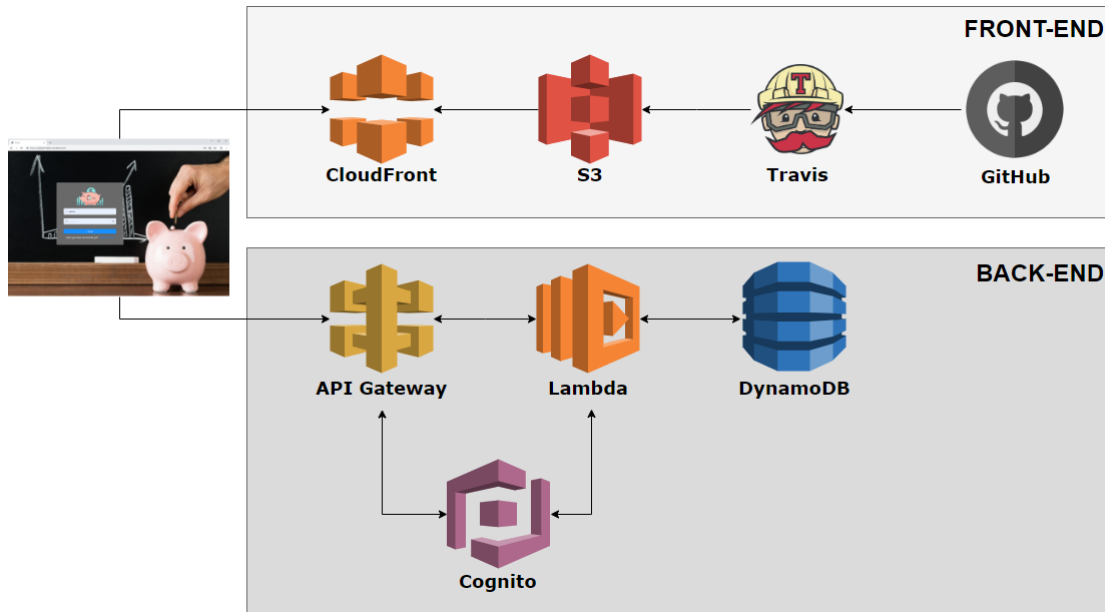


Figura 13.1: Diagrama de les tecnologies utilitzades per la realització del sistema [13]

13.1 Arquitectura de Software del sistema

Pel que fa a l'arquitectura de software del sistema, a la imatge 13.2 es representen les classes del sistema en format UML, un diagrama de classes on apareixen les principals classes del nostre sistema. Per explicar les classes principals cal començar a llegir el diagrama per la classe usuari, un usuari pot tenir (o no tenir) contactes i transaccions i pot pertànyer a diferents grups, però sempre tindrà un saldo (que indica els diners actuals de l'usuari). També cal destacar que les transaccions sempre tindran una categoria, és per això que la relació amb la subclasse és *complete* i sempre serà una de les dues categories especificades (Income o Outcome) per això és *disjoint*, només pot tenir una de les dues categories. Pel que fa als grups, aquest poden o no tenir moviments, i aquests finalment estan relacionats amb els usuaris, ja que un moviment afecta 2 o més usuaris.

Les restriccions textuais pel diagrama de la imatge 13.2 son les següents:

1. Claus Externes:
 - (a) Usuari - username
 - (b) Saldo - balanceID
 - (c) Grups - groupID
 - (d) Transacció - userID + transactionID
 - (e) Moviments - movementID
2. Un usuari no pot tenir accés al e-mail dels seus contactes.
3. Les data d'una transacció han de ser anterior a la data actual.
4. Les data d'un moviment han de ser anterior a la data actual.
5. El *amount* d'un moviment sempre ha de ser un valor positiu.
6. L'usuari que ha pagat un moviment no ha d'apareixer a la llista de participants.
7. Un usuari no pot ser contacte de si mateix.

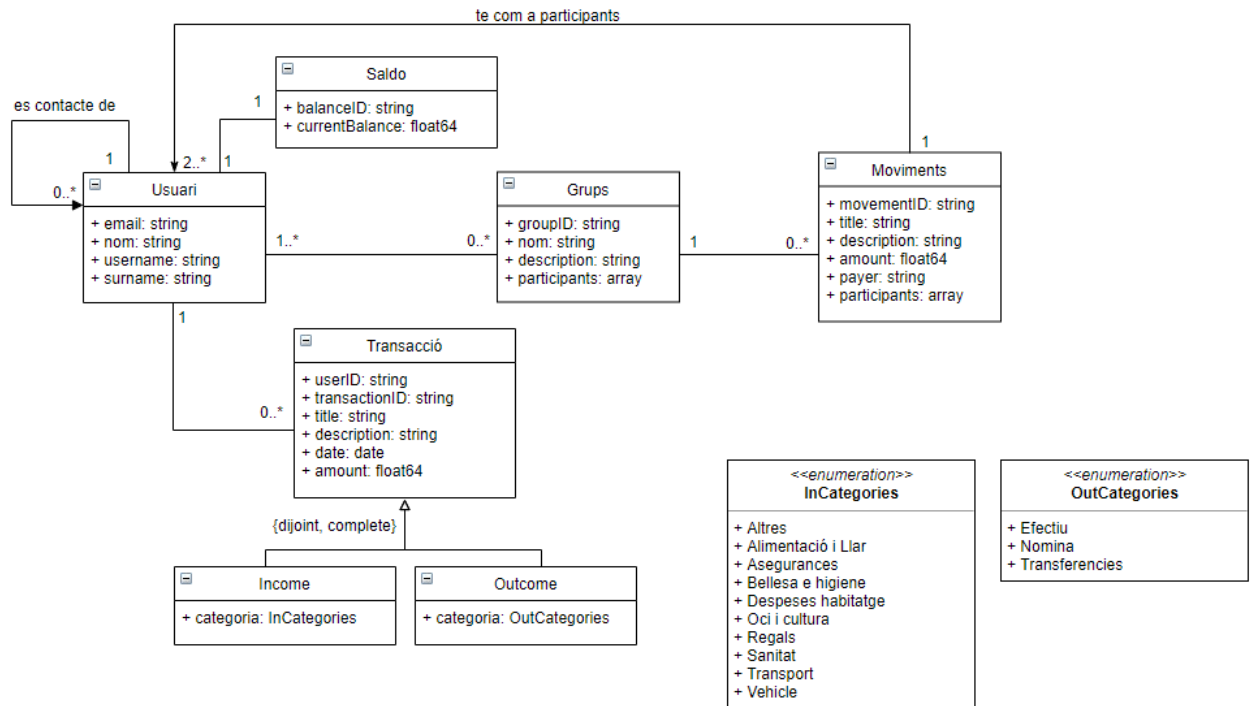


Figura 13.2: Diagrama de classes utilitzat dins el sistema [13]

13.2 Patrons de disseny

En aquest apartat s'expliquen els patrons de disseny utilitzats tant pel Back-End com pel Front-End en el desenvolupament del sistema i com s'han utilitzat.

Patrons de disseny utilitzats pel Back-End

Pel Back-End s'ha utilitzat una arquitectura Serverless on practicamente tot està controlat per els sistemes d'AWS, per això s'ha investigat quins son els patrons de disseny d'aquesta plataforma i quin s'adecua al nostre sistema [14]. El patró que més s'assembla al nostre sistema es el de *The Gatekeeper*, tot i que no es 100% igual ja que el nostre sistema té l'Autenticació a través de Cognito i no de Lambdas.

Patrons de disseny utilitzats pel Front-End

Pel que fa al Front-End s'han utilitzat diversos patrons de disseny, per tal de trobar exemples d'aquests s'han buscat recursos web [15].

S'ha utilitzat el patró **Constructor** (imatge 13.3) [16]. En el nostre cas s'ha utilitzat en diversos casos, per exemple pel cas de les cookies, es construeixen cookies noves cada cop que s'inicialitza el sistema. Hi ha més casos de l'ús d'aquest patró, però aquest és el cas més clar.

```

1  function Car( model, year, miles ) {
2
3      this.model = model;
4      this.year = year;
5      this.miles = miles;
6
7      this.toString = function () {
8          return this.model + " has done " + this.miles + " miles";
9      };
10 }
11
12 // Usage:
13
14 // We can create new instances of the car
15 var civic = new Car( "Honda Civic", 2009, 20000 );
16 var mondeo = new Car( "Ford Mondeo", 2010, 5000 );
17
18 // and then open our browser console to view the
19 // output of the toString() method being called on
20 // these objects
21 console.log( civic.toString() );
22 console.log( mondeo.toString() );

```

Figura 13.3: Exemple del patró constructor

S'ha utilitzat el patró **Observador** (imatge 13.4) [17] per casos de modificació de propietats, aquest patró s'utilitza al nostre sistema per exemple en el cas de les taules dinàmiques, aquestes varien en funció si varia el paràmetre que conté les dades dins l'estat i aquest es modifica en el cas de realitzar les funcions d'editar, crear, esborrar...

```

1  // The Observer
2  function Observer(){
3      this.update = function(){
4          // ...
5      };
6  }

```

Figura 13.4: Exemple simple del patró observador

S'ha utilitzat el patró **Prototip** (imatge 13.5) [18] pel cas de la creació de modals o de taules, ja que s'ha utilitzat una implementació per diferents versions d'aquests. A la imatge d'exemple es veu l'ús del patró en una variable, en el nostre cas està més utilitzat al rendering dels diferents components.


```

1  var vehiclePrototype = {
2
3      init: function ( carModel ) {
4          this.model = carModel;
5      },
6
7      getModel: function () {
8          console.log( "The model of this vehicle is.." + this.model);
9      }
10 };
11
12
13 function vehicle( model ) {
14
15     function F() {};
dieciséis  F.prototype = vehiclePrototype;
17
18     var f = new F();
19
20     f.init( model );
21     return f;
22 }
23
24
25 var car = vehicle( "Ford Escort" );
26 car.getModel();

```

Figura 13.5: Exemple amb el cas de variables del patró Prototip

També s'han utilitzat de manera indirecta a través de l'ús de llibreries patrons com per exemple el patró **fatxada** (imatge 13.6) [19] que mentre el codi emmagatzema molta informació, a la pantalla no apareixen totes les dades. Per exemple en el cas de les taules aquestes també tenen l'identificador guardat tot i que no es mostra per pantalla.

```

1  var addMyEvent = function( el,ev,fn ){
2
3      if( el.addEventListener ){
4          el.addEventListener( ev,fn, false );
5      }else if(el.attachEvent){
6          el.attachEvent( "on" + ev, fn );
7      } else{
8          el["on" + ev] = fn;
9      }
10
11 };

```

Figura 13.6: Exemple simple del patró fatxada

Per últim destacar l'ús del patró **factoria** (imatge 13.7) [20] a l'hora de crear els diferents components de les interfícies tot i que no ha estat implementat sinó que s'ha utilitzat les versions pròpies de les llibreries.

Approach #1: Modify a VehicleFactory instance to use the Truck class

```

1  var movingTruck = carFactory.createVehicle( {
2      vehicleType: "truck",
3      state: "like new",
4      color: "red",
5      wheelSize: "small" } );
6
7  // Test to confirm our truck was created with the vehicleClass/prototype Truck
8
9  // Outputs: true
10 console.log( movingTruck instanceof Truck );
11
12 // Outputs: Truck object of color "red", a "like new" state
13 // and a "small" wheelSize
14 console.log( movingTruck );

```

Figura 13.7: Exemple simple d'ús del patró factoria

13.3 Esquema de les bases de dades

Per aquest sistema s'ha decidit utilitzar les bases de dades de DynamoDB (com s'ha explicat anteriorment) aquesta base de dades és una base de dades no relacional que consta de col·leccions, per tant totes les relacions entre les dades es fan en desenvolupar l'API i agafar les dades de les diferents col·leccions, les relacions entre les dades són si fa o no fa el mateix que s'ha presentat al diagrama de classes del sistema (figura 13.2)

El sistema consta de 7 col·leccions:

1. **Usuaris:** Col·lecció encarregada de guardar les dades bàsiques dels usuaris (les dades importants d'aquests estan emmagatzemades a Cognito) és per aquest motiu que a la col·lecció només tenim el nom d'usuari, e-mail, nom i cognom de l'usuari. La clau primària de la taula és el nom d'usuari, no poden haver-hi dos usuaris amb el mateix nom d'usuari, però sí que pot passar que tinguin el mateix e-mail.

```

Usuaris: {
  username: string,
  email: string,
  completeName: string,
  surname: string
}

```

2. **Balance:** Col·lecció encarregada de guardar el saldo actual de l'usuari, aquesta taula té com a clau primària l'identificador del balanç que correspon al nom d'usuari per tal de saber a quin usuari pertany el balanç i només té un paràmetre, el saldo actual de l'usuari.

```

Balance: {
  balanceID: string,
  currentBalance: float64
}

```

3. **Transaccions:** Col·lecció encarregada de guardar les transaccions dels usuaris, aquesta té com a paràmetres l'identificador de l'usuari, l'identificador de la transacció, la quantitat, la categoria, la data, la descripció, si és despesa o no i el títol. Les claus primàries de la taula són l'identificador de l'usuari (per tal d'identificar a quin usuari pertany la transacció) i l'identificador de la transacció (per tal de diferenciar les diferents transaccions de l'usuari).

```

Transaccions: {
  userID: string,
  transactionID: string,

```

```
    amount: float64,  
    category: string,  
    date: string,  
    description: string,  
    expense: boolean,  
    title: string  
}
```

4. **Contactes:** Col·lecció encarregada de guardar les dades dels contactes, els dos caps que emmagatzema són les claus primàries, ja que no importa si l'usuari està al camp d'identificador d'usuari o identificador de contacte (en els dos camps l'identificador correspon al nom d'usuari), en els dos casos apareixerà el contacte pertinent.

```
Contactes: {  
    userID: string,  
    contactID: string  
}
```

5. **Grups:** Col·lecció encarregada de guardar les dades dels grups, aquesta té com a paràmetres l'identificador del grup, una descripció, el nom del grup i els participants. La clau primària del grup és l'identificador únic del paràmetre identificador del grup.

```
Grups: {  
    groupID: string,  
    description: string,  
    name: string,  
    participants: array[string]  
}
```

6. **GrupMovements:** Col·lecció encarregada de guardar les dades dels moviments dels grups, aquesta té com a paràmetres l'identificador del moviment, l'identificador del grup, la quantitat, la data, una descripció, la llista de participants, el pagador i un títol. Les claus primàries de la col·lecció són l'identificador del moviment (generat com a identificador únic) i l'identificador del grup (per identificar a quin grup pertany la transacció).

```
GrupMovements: {  
    movementID: string,  
    groupID: string,  
    amount: float64,  
    date: string,  
    description: string,  
    participants: array[string],  
    payer: string,  
    title: string  
}
```

7. **UserMovements:** Col·lecció encarregada de guardar les relacions de pagament de moviments entre els usuaris d'un grup, aquesta té com a paràmetres l'identificador del moviment, l'identificador de l'usuari, la quantitat, si està pagat o no i el pagador. Les claus primàries de la col·lecció són l'identificador del moviment i l'identificador de l'usuari (que correspon al nom d'usuari d'aquest que deu diners).

```
UserMovements: {  
    movementID: string,  
    userID: string,  
    amount: float64,  
    payed: bool,  
    payer: string  
}
```

13.4 Disseny de les interfícies - Funcionalitats aplicació

En aquest subapartat s'explica el funcionament de l'aplicació i el seu disseny. L'objectiu és explicar totes les funcionalitats disponibles (totes les imatges presentades a continuació pertanyen a la versió web del sistema).

Pel que fa al disseny de l'aplicació s'ha decidit utilitzar un disseny minimalista i utilitzant com a paleta de colors els tons blaus corresponent als codis hexadecimals #40a9ff (per als fons foscos) i #003495 (per als fons clars). Tots els detalls de disseny es poden veure a les imatges que es presenten a continuació amb l'explicació de les funcionalitats.

13.4.1 Procés de login i logout

En iniciar l'aplicació l'usuari accedeix a la pantalla de Login (imatge 13.8) on ha d'introduir el nom d'usuari i la contrasenya, al prémer el botó de *Log In* es consulten les dades al sistema d'usuaris, en cas d'introduir les dades de manera incorrecta, s'avisava a l'usuari del fet que les dades són incorrectes, en cas d'introduir les dades de manera correcta s'accedirà a la pantalla principal de l'aplicació (explicada més endavant).

En el cas que no disposi d'un compte, s'inicia el procés de registre (explicat al punt següent) mitjançant el botó de *Register Now!*

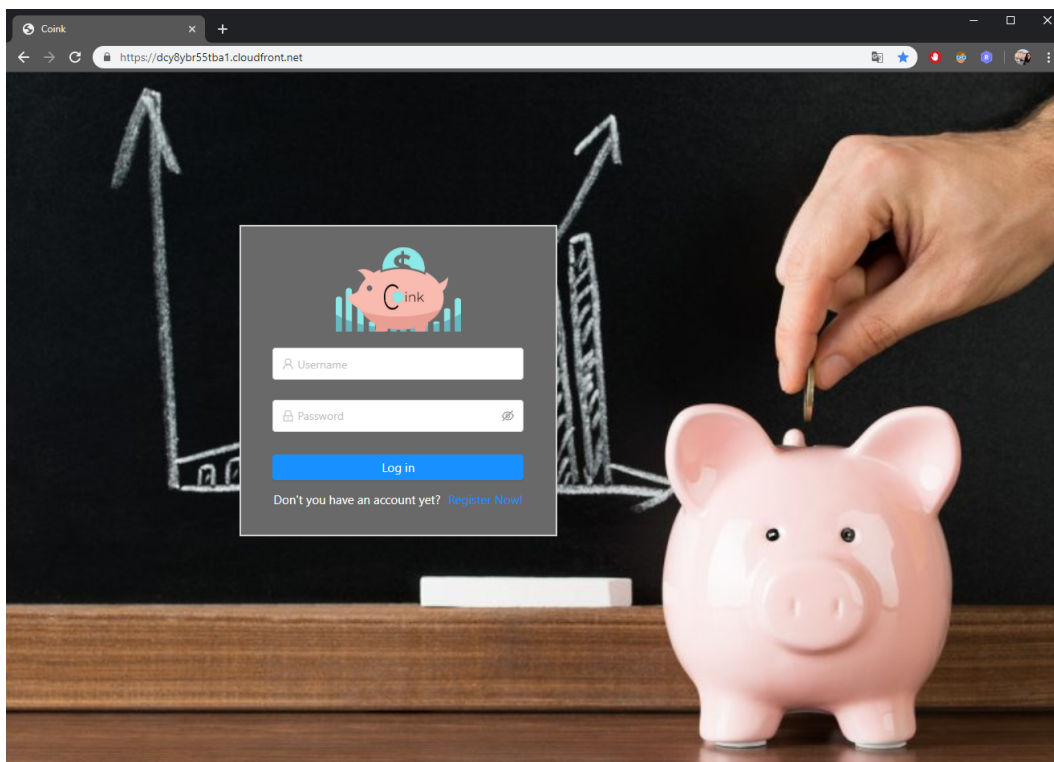


Figura 13.8: Pantalla inicial de l'aplicació - login

Per tal de fer logout de l'aplicació caldrà premer el botó de Logout situat a la banda dreta del menú superior de la pantalla (imatge 13.9).



Figura 13.9: Menú superior de l'aplicació

13.4.2 Procés de registre

El procés de registre consta d'un seguit de 4 pantalles on es demanen diferents dades a cadascuna, a la primera pantalla (imatge 13.10), que és la que apareix en prémer el botó de *Register Now!* de la pantalla anterior (login) s'ha d'introduir un nom d'usuari, un e-mail vàlid, i la contrasenya dos cops per tal de fer la comprovació que s'ha escrit de manera correcta (la contrasenya ha de seguir els criteris de: mida de 8 caràcters o més i ha de contenir una majúscula, una minúscula, un número i un símbol) en aquesta pantalla també hi ha l'opció d'anar enrere, el que et portarà a la pantalla de login.

Un cop introduïdes totes les dades quan es prem el botó de *Register*, si les dades són correctes, és a dir que el nom d'usuari conté una única paraula, que no hi ha cap usuari donat d'alta amb aquell nom d'usuari i la contrasenya segueix el criteri, es passarà a la segona pantalla del procés, en cas contrari s'informarà l'usuari de què les dades són incorrectes.

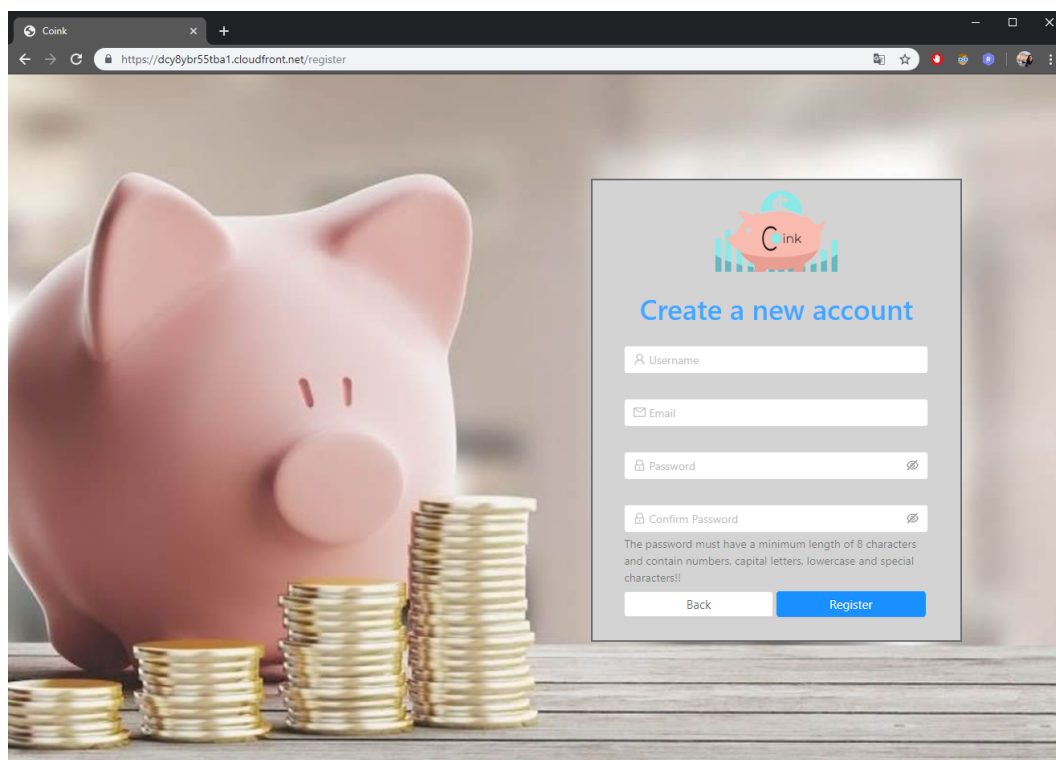


Figura 13.10: Primera pantalla del procés de registre

Un cop a la segona pantalla del procés (imatge 13.12) l'usuari ha d'haver rebut a l'e-mail (imatge 13.11) que ha indicat a la pantalla anterior un codi de verificació, en el cas de no rebre el correu es pot clicar al botó de *Resend validation code* per tal de tornar a enviar el correu, un cop introduït el codi i quan l'usuari clica el botó de *Validate*, si les dades no són correctes, s'avisarà a l'usuari de què les dades introduïdes no són correctes, en cas contrari es passarà a la tercera pantalla del procés de registre.

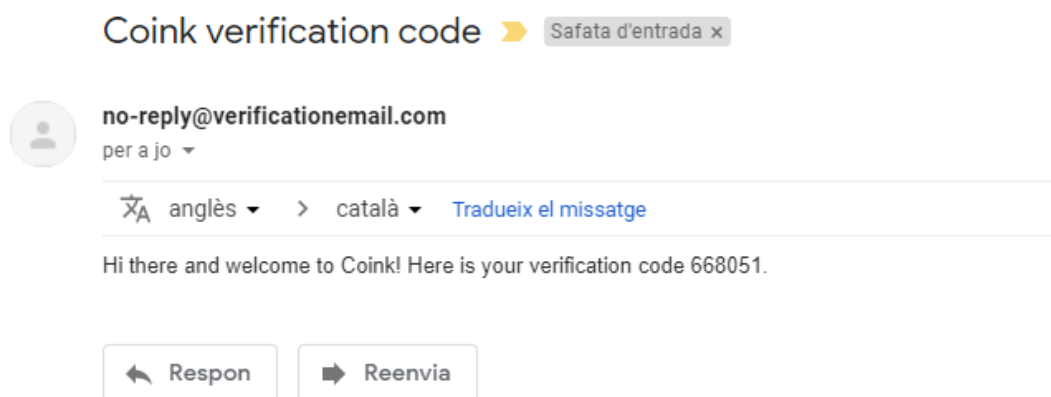


Figura 13.11: Exemple de email de verificació de compte

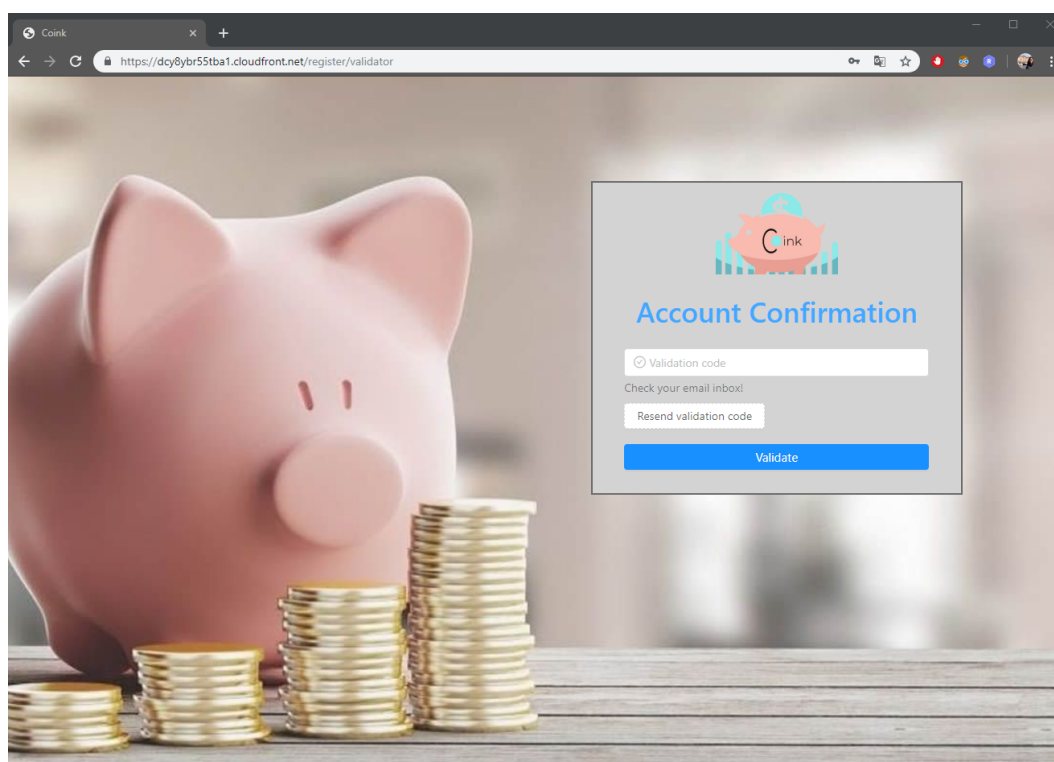


Figura 13.12: Segona pantalla del procés de registre

Un cop a la tercera pantalla del procés (imatge 13.13) l'usuari ha estat validat i es demana que es faci login, per no haver de tornar a la pantalla de login i com el nom d'usuari ja està emmagatzemat al sistema, es demana només introduir un altre cop la contrasenya com a mesura de seguretat, en cas d'introduir la contrasenya de manera correcta s'accedeix a la quarta i última pantalla del procés, en cas contrari, s'avisava a l'usuari de què les dades introduïdes no són correctes.

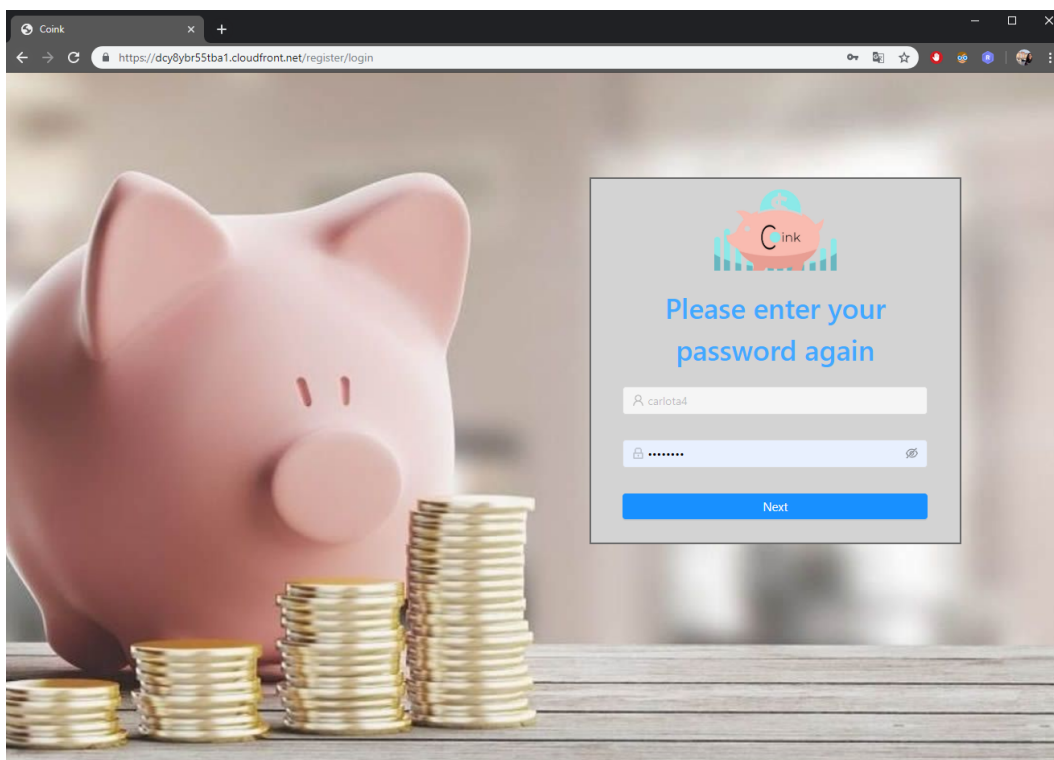


Figura 13.13: Tercera pantalla del procés de registre

Un cop a la quarta i última pantalla del procés (imatge 13.14) l'usuari ja està logejat al sistema i es demana que introdueixi el *Current Balance*, és a dir el saldo que té actualment per tal de no començar amb 0€ i haver d'introduir de manera manual com a transacció el saldo, en aquest camp només es permet l'entrada de números, tant positius com negatius (ja que l'usuari es pot quedar amb un balanç negatiu). Un cop es prem el botó de *Start!!* s'accedeix a la pantalla principal de l'aplicació.

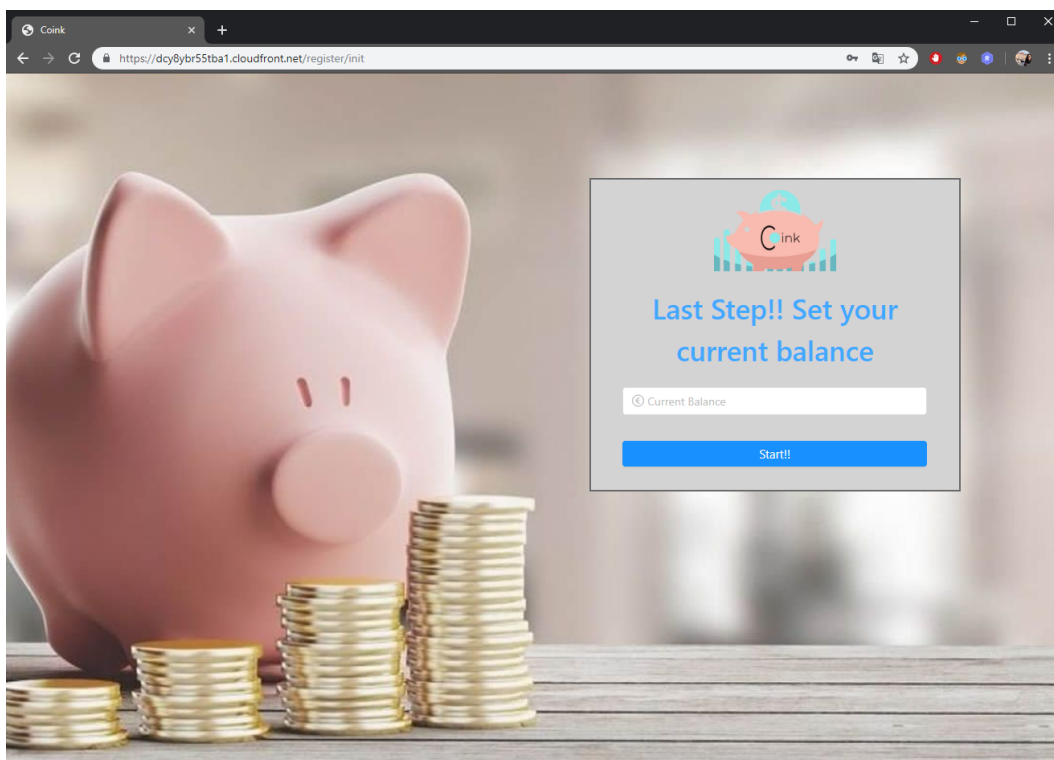


Figura 13.14: Quarta pantalla del procés de registre

13.4.3 Pantalla principal

La pantalla principal de l'aplicació és únicament de consulta, a part que podem veure a la imatge 13.15 es pot consultar el saldo actual de l'usuari que s'anirà actualitzant un cop es fan les diferents transaccions, les estadístiques dels grups, el total que l'usuari deu i el total que li deuen i per últim les estadístiques de l'any, de manera numèrica el que s'ha gastat i el que ha guanyat i de forma gràfica (encara que no es veu a la imatge, són com les de la imatge 13.25) el que ha gastat/guanyat en cada categoria establerta. Pel que fa a la part superior de la pantalla, apareix una frase inspiradora per tal d'incentivar a l'usuari a estalviar (imatge 13.16).

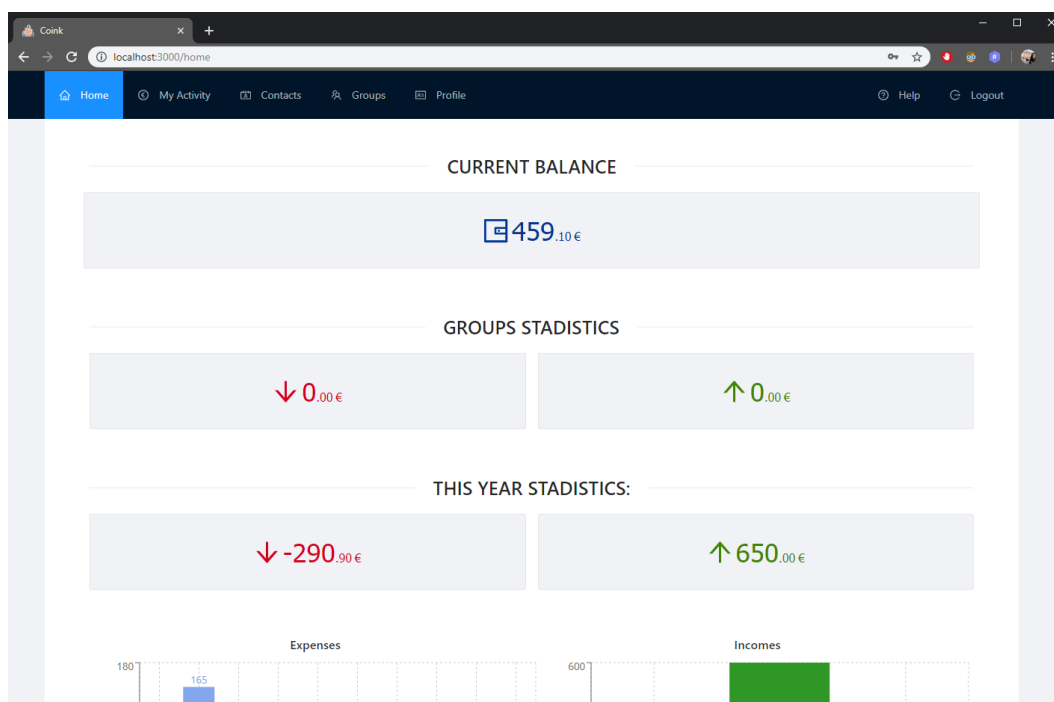


Figura 13.15: Part de la pantalla principal de l'aplicació



Figura 13.16: Exemple d'una de les frases inspiradores [36]

Activitat

La pantalla d'activitat de l'usuari hi ha dues grans seccions, l'apartat de totes les transaccions i l'apartat de les transaccions per any.

A l'apartat de totes les transaccions (imatge 13.17) apareix una taula amb totes les transaccions de l'usuari, en aquesta taula es pot ordenar per qualsevol dels camps (excepte pel tipus) i també es pot filtrar per tots ells (excepte per la quantitat). En aquesta pantalla es poden realitzar tota mena de funcionalitats sobre les transaccions, es pot exportar en Excel (imatge 13.22) prement el botó *Export Excel Table*, es pot afegir una nova entrada (imatge 13.18) prement el botó de *Add Income* o una nova sortida (imatge 13.19) prement el botó de *Add Expense*, també es poden editar (imatge 13.20) prement el botó del llapis dins d'una transacció o esborrar (imatge 13.21) prement el botó de la brossa.

€ MY ACTIVITY

↑ Add Income ↓ Add Expense Export Excel Table

All my transactions Year transaction

Date	Title	Description	Category	Amount (€)	Type
14/06/2019	Sopar fora	Sopar al vapiano	Alimentacio i Llar	-15	EXPENSE
13/06/2019	Gasolina	Omplir el diposit de gasolina	Vehicle	-60	EXPENSE
12/06/2019	Privalia	Roba d'estiu	Bellesa e higiene	-60	EXPENSE
28/05/2019	Nomina	Nomina Maig HP	Nomina	600	INCOME
14/05/2019	Cine	Entrada cine	Oci i cultura	-5.9	EXPENSE
07/05/2019	Diners cumple	Diners avis pel meu regal de cumple	Efectiu	50	INCOME
01/05/2019	Super	Compra del mes	Alimentacio i Llar	-150	EXPENSE

Figura 13.17: Pantalla on apareixen totes les transaccions de l'usuari

Per tal de crear una transacció, tant entrada (imatge 13.18) com sortida (imatge 13.19) és necessari omplir tots els camps, en cas contrari el sistema no permetrà la creació, un cop creada, s'afegeix automàticament a la taula. No es poden afegir categories no especificades, dates futures o quantitats no numèriques, per una sortida la quantitat ha de ser negativa mentre que per una entrada sempre serà positiva.

New Income

* DatePicker
2019-06-22

* Title
Enter a title

* Description
Enter a description

* Category
Select a category

* Amount
Amount

Cancel Create

New Expense

* DatePicker
2019-06-22

* Title
Enter a title

* Description
Enter a description

* Category
Select a category

* Amount
Amount

Cancel Create

Figura 13.18: Modal per afegir una entrada Figura 13.19: Modal per afegir una sortida

Una transacció no és 100% editable (imatge 13.20), els camps editables són el títol, la descripció, la categoria i la quantitat, com s'ha indicat als requisits, un cop introduïdes les dades correctament a tots els camps si es prem el botó del *disquet* (guardar) es realitzarà el canvi intern per tal de tenir les dades actualitzades, en el cas de prémer la *creu vermella* s'esborraran els canvis realitzats.

Date	Title	Description	Category	Amount (€)	Type	
14/06/2019	Sopar fora	Sopar al vapiano	Alimentacio i Llar	-15.00	EXPENSE	
13/06/2019	Gasolina	Omplir el diposit de gasolina	Vehicle	-60	EXPENSE	

Figura 13.20: Fragment de la taula de transaccions on es veu una transacció editable

En esborrar una transacció del sistema (imatge 13.21) no es pot tornar enrerre així que en el cas de prémer *OK* la transacció s'esborrarà per sempre, en el cas de voler-la recuperar l'única manera és tornant-la a crear, si es prem el botó de *cancelar*, no es fa cap canvi al sistema.

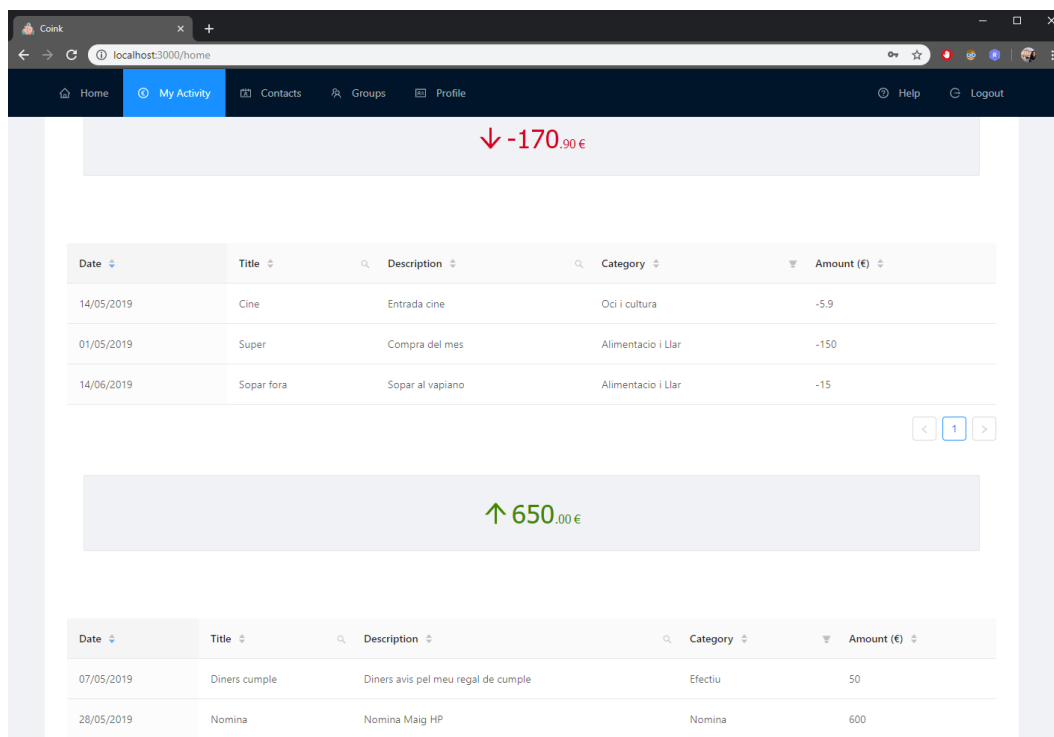
Date	Title	Description	Category	Amount (€)	Type	
14/06/2019	Sopar fora	Sopar al vapiano	Alimentacio i Llar	-15	EXPENSE	

Figura 13.21: Pantalla on es veu una transacció anant a ser esborrada

	A	B	C	D	E	F	G
1	Date	Titol	Descripció	Categoria	Tipus	Quantitat	
2	07/05/2019	Diners cumple	Diners avis pel meu regal de cumple	Efectiu	Income	50	
3	28/05/2019	Nomina	Nomina Maig HP	Nomina	Income	600	
4	14/05/2019	Cine	Entrada cine	Oci i cultura	Expense	-5.9	
5	01/05/2019	Super	Compra del mes	Alimentacio i Llar	Expense	-150	
6	14/06/2019	Sopar fora	Sopar al vapiano	Alimentacio i Llar	Expense	-15	
7							

Figura 13.22: Template de l'excel que s'exporta al premer el botó de exportar

L'apartat de transaccions per any és únicament de consulta, com podem veure a la imatge 13.23 apareixen de manera separada les entrades i les sortides amb unes taules que igual que a l'apartat comentat anteriorment, podem filtrar i ordenar per les diferents columnes, també hi ha un número a sobre de cada taula que ens mostra el total de les sortides (en vermell) i de les entrades (en verd) que hi ha hagut a l'any seleccionat.



The screenshot shows a web application interface for 'Coink'. The 'My Activity' tab is selected. At the top, a summary bar shows a red downward arrow and '-170.90 €'. Below this is a table of transactions:

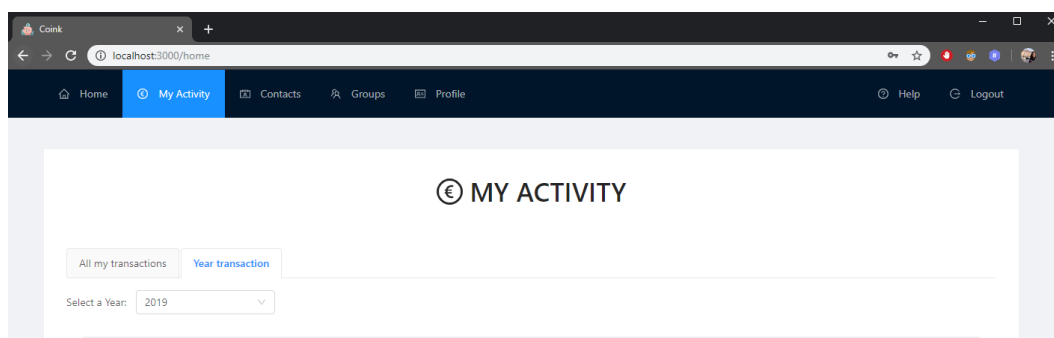
Date	Title	Description	Category	Amount (€)
14/05/2019	Cine	Entrada cine	Oci i cultura	-5.9
01/05/2019	Super	Compra del mes	Alimentacio i Llar	-150
14/06/2019	Sopar fora	Sopar al vapiano	Alimentacio i Llar	-15

Below the table is a green upward arrow and '650.00 €'. Another table of transactions follows:

Date	Title	Description	Category	Amount (€)
07/05/2019	Diners cumple	Diners avis pel meu regal de cumple	Efectiu	50
28/05/2019	Nomina	Nomina Maig HP	Nomina	600

Figura 13.23: Part de la pantalla que mostra les transaccions per any

Per tal de seleccionar l'any del que es vol consultar, a la part superior d'aquesta pantalla (imatge 13.24), hi ha un dropdown per tal de seleccionar l'any, en canviar l'any, totes les dades canvien i es mostra l'any seleccionat.



The screenshot shows the top part of the 'My Activity' page. It features a header with '€ MY ACTIVITY'. Below the header, there are two tabs: 'All my transactions' and 'Year transaction' (which is selected). Under the 'Year transaction' tab, there is a 'Select a Year:' label followed by a dropdown menu showing '2019'.

Figura 13.24: Part superior de la pantalla que mostra les transaccions per any

A la part inferior de la pantalla (i de la mateixa manera que a la pantalla principal) trobem les estadístiques que mostra per cada categoria quan s'ha gastat l'usuari (imatge 13.25).

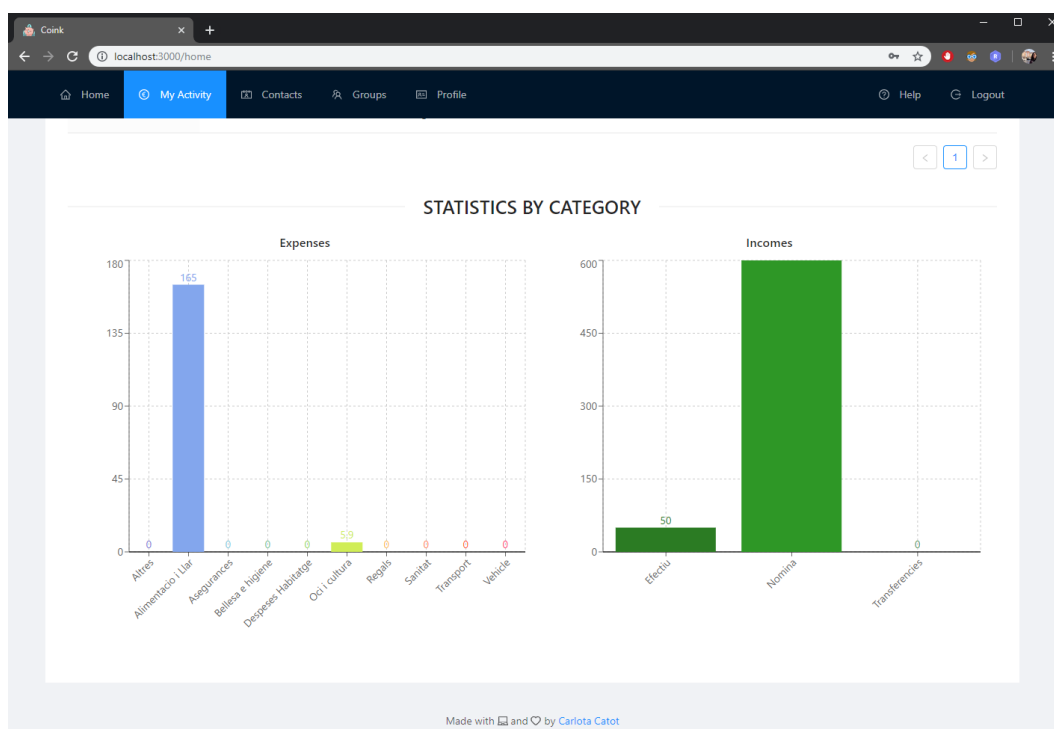


Figura 13.25: Gràfiques que apareixen a la part inferior de la pantalla de transaccions per any i de la pantalla principal

13.4.4 Contactes

La pantalla de contactes de l'usuari (imatge 13.26) és una pantalla molt simple, aquí l'usuari pot consultar la informació dels seus contactes i els deutes que té amb cadascun, també pot afegir nous contactes prement el botó d'*Add Contact* o esborrar els usuaris que ja té prement el botó de la *paperera* que hi ha per cada contacte.

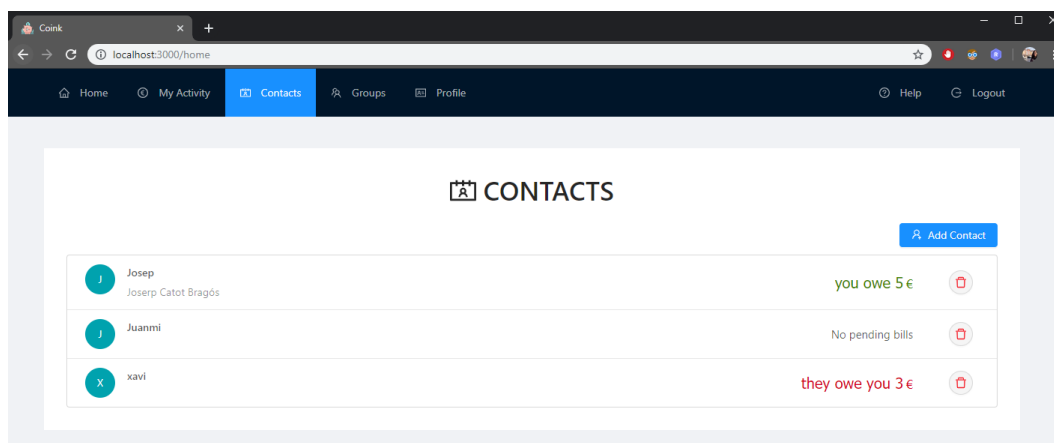


Figura 13.26: Pantalla de contactes de l'usuari

Per tal d'afegir un nou contacte (imatge 13.27) és necessari introduir un nom d'usuari o seleccionar-lo de la llista que apareix en forma d'autocompletar quan l'usuari comença a escriure. En el cas d'intentar afegir un usuari que no estigui a la llista, s'avisarà a l'usuari de què l'usuari que s'intenta afegir no és vàlid perquè no pertany al sistema, és a dir, que només accepten usuaris de la llista que apareix un cop l'usuari comença a escriure.

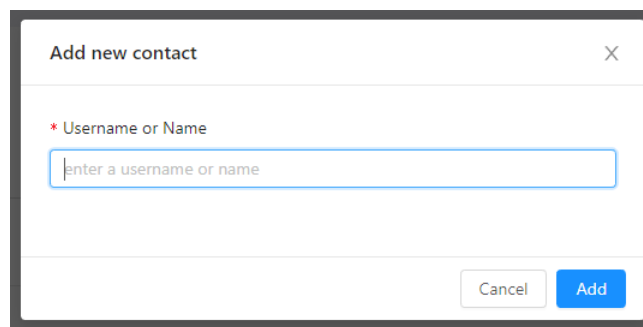


Figura 13.27: Modal que permet afegir un contacte

13.4.5 Grups

La pantalla de grups de l'usuari (imatge 13.28) permet realitzar diferents funcionalitats. A la part superior de la pantalla apareix un resum dels deutes de tots els grups als quals pertany l'usuari, i també un balanç d'aquest. Just a sota podem veure una llista dels diferents grups als quals pertany l'usuari, per cada grup apareix els deutes que té amb cada usuari amb l'opció de *Settle debts*, que posarà el deute amb l'usuari a 0 i afegirà un moviment indicant la transacció realitzada. Dins de cada grup trobem dos botons, el d'esborrar que funciona de la mateixa manera que pels contactes o les transaccions i el de *See all movements* que obrirà una nova finestra amb la informació del grup. Finalment, el botó de *Create Group* permet crear nous grups.

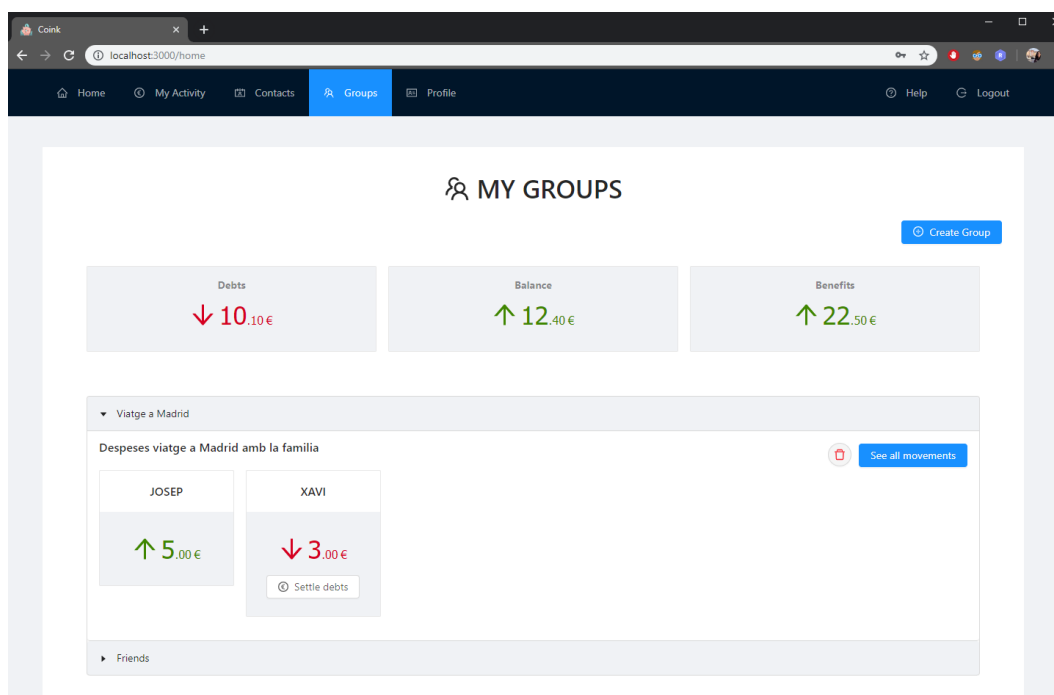


Figura 13.28: Pantalla de grups de l'usuari

Per tal de crear un nou grup (imatge 13.29) és necessari introduir el nom del grup, una descripció i algun participant, és necessari afegir com a mínim un participant i l'usuari que està creant el grup sempre formarà part dels grups que hagi creat. A l'hora d'afegir un participant només serà possible afegir usuaris de la llista de contactes. Un cop introduïdes totes les dades es clica el botó de *Create* i el grup apareixerà a la pantalla explicada abans, si les dades eren correctes.

Figura 13.29: Modal que permet crear un nou grup

Un cop s'ha clicat el botó de *See all movements* s'obre una nova finestra al navegador i apareix la pantalla d'informació del grup, en aquesta pantalla disposa de tres seccions.

A la primera secció (imatge 13.30) apareix una taula amb tots els moviments realitzats al grup per tots els usuaris, en aquesta taula es pot ordenar per qualsevol dels camps i també es pot filtrar per tots ells (excepte per la quantitat). En aquesta pantalla es poden realitzar tota mena de funcionalitats sobre els moviments del grup, es pot exportar en Excel, prement el botó *Export Excel Table* (mateixa funcionalitat que per les transaccions), es pot afegir un nou moviment prement el botó de *Add Movement*, també es poden editar prement el botó del llapis dins d'una transacció o esborrar prement el botó de la broixa, no es presenten imatges de les funcionalitats, ja que visualment són iguals que per les transaccions.

Date	Title	Description	Payer	Participants	Amount (€)	
09/06/2019	Bus	Tiquet de bus per arribar a l'estació	cristina	carlota4	10	
09/06/2019	Botiga Aeroport	Menjar per l'avió	cristina	josep - carlota4	15	
08/06/2019	Souvenirs	Botiga de souvenirs	Josep	carlota4	30	
08/06/2019	Brenar	Chocolata amb churros a la plaça	xavi	josep - carlota4	21	
07/06/2019	Museu d'història natural	Pagament de les entrades per entrar al museu	carlota4	xavi - cristina - Josep	40	
07/06/2019	Sopar	Pagament del sopar	cristina	xavi	30	

Figura 13.30: Pantalla d'informació d'un grup - Tots els moviments del grup

A la segona secció apareix una llista de participants molt semblant a la llista de grups de la pantalla inicial dels grups, en aquest cas cada element de la llista pertany a un participant del grup. En el cas de seleccionar el nom d'usuari de l'usuari que està utilitzant l'aplicació, apareix el botó de *Settle debts*, que posarà el deute amb l'usuari a 0 i afegirà un moviment

indicant la transacció realitzada.

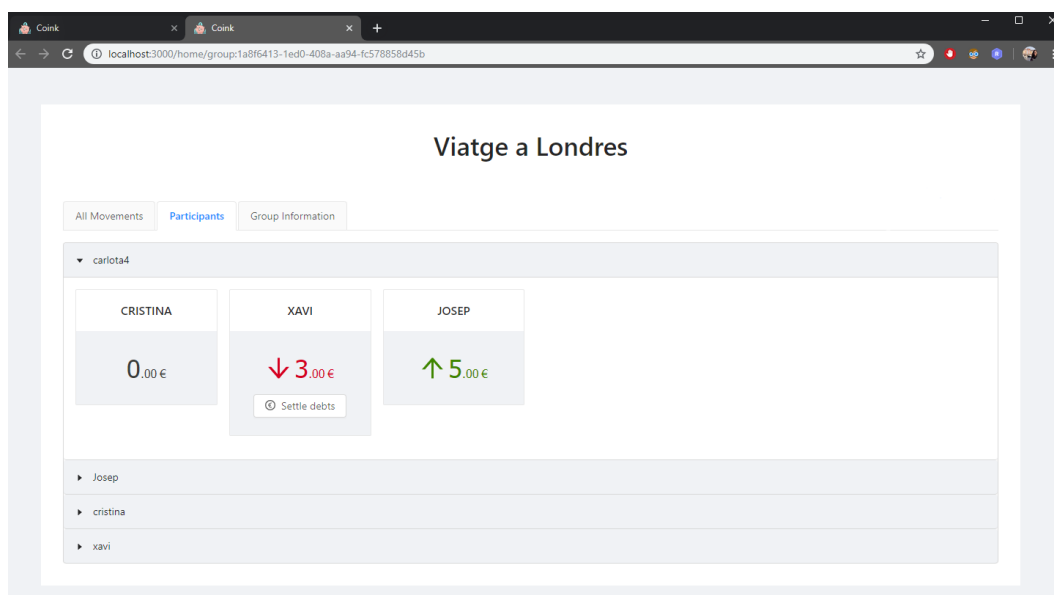


Figura 13.31: Pantalla d'informació d'un grup - Resum participants

A la tercera i última secció apareix un formulari on es pot editar la informació actual del grup, tots els camps són editables i un cop realitzats els canvis aquests es guarden apretant el botó de *Save Changes*.

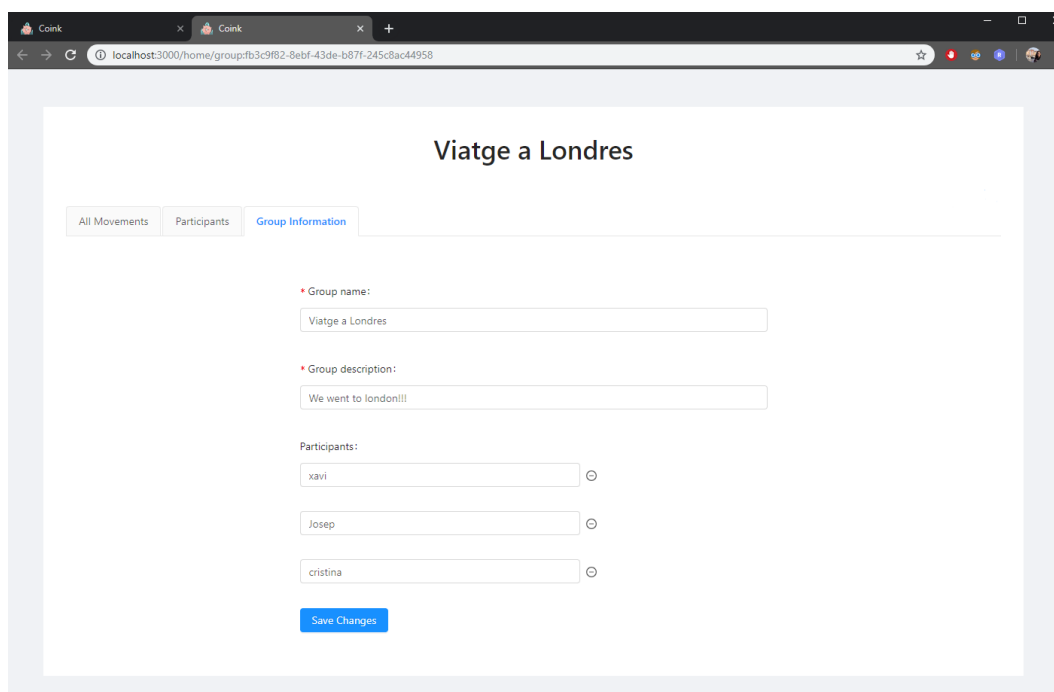


Figura 13.32: Pantalla d'informació d'un grup - Editar grup

13.4.6 Perfil

A la pantalla de perfil de l'usuari (imatge 13.33) es pot consultar i modificar la informació de l'usuari actual, les dades que corresponen al nom d'usuari i a l'e-mail no són editables, però el nom i el cognom es poden editar en qualsevol moment inserint el que es vol posar en el camp editable i prement el botó de *Save Changes*. També es pot editar la contrasenya prement el botó de *Change password*.

The screenshot shows a web browser window with the URL 'localhost:3000/home'. The application has a dark blue navigation bar with links: Home, My Activity, Contacts, Groups, Profile (active), Help, and Logout. The main content area is titled 'MY PROFILE' and contains the following fields:

- Username:** CARLOTA4
- E-mail:** cakbra@gmail.com
- Name:** Carlota
- Surname:** Catot Bragós

There is a 'Change Password' button in the top right of the profile section and a 'Save Changes' button at the bottom.

Figura 13.33: Pantalla d'informació de l'usuari

Quan l'usuari prem el botó de *Change password* s'obre un el modal de la imatge 13.34 on l'usuari ha d'introduir la contrasenya actual i la nova contrasenya que es vol posar, per assegurar que la nova contrasenya s'està escrivint de manera correcta, s'ha d'introduir dos cops. En cas d'introduir les dades de manera correcta, és a dir, correctament la contrasenya actual, apareixerà un missatge informant l'usuari de què les dades han estat modificades correctament, en cas contrari s'avisarà a l'usuari de què les dades eren incorrectes.

The 'Change Password' modal has a title bar with a close button (X). It contains the following fields and text:

- * Old Password:** Password
- * New Password:** Password
- * Confirm New Password:** Password

Below the fields, a note reads: "Remember that the password must have a minimum length of 8 characters and contain numbers, capital letters, lowercase and special characters!!". At the bottom right, there are 'Cancel' and 'Change' buttons.

Figura 13.34: Modal que permet canviar la contrasenya de l'usuari

Capítol 14

Diari d'execució

A continuació s'explicarà en profunditat els diferents Sprints que s'han seguit, no tots ells han tingut la mateixa durada per diferents factors que també s'explicaran a continuació. Per cada sprint es presentarà el Backlog que hi havia abans de començar l'Sprint (amb totes les històries d'usuari plantejades) i el panell de l'Sprint on es pot veure per cada història d'usuari les tasques a realitzar (a les imatges dels panells dels Sprints s'han col·locat les targetes de manera que es vegin totes a la pantalla més enllà del punt en el qual es trobin).

14.1 Sprint 1

Donat el backlog inicial del projecte (figura 14.1), pel primer Sprint s'ha decidit assignar les històries d'usuari **Set Up del Sistema - 16 punts** i **Session Management - 3 punts** i una durada d'una setmana (del 04 de març del 2019 a l'11 de març del 2019), ja que les tasques a realitzar (figura 14.2) són més de disseny de l'arquitectura i de familiarització i aprenentatge amb el sistema d'AWS, per tal de definir des del primer moment una bona arquitectura del sistema. Durant aquest Sprint s'estan realitzant paral·lelament tasques de documentació durant l'assignatura de GEP. Tota la informació de l'Sprint es pot consultar al taiga [21].

Tasques de Front-End

1. **React SetUp:** Creació del sistema de React juntament amb el component de disseny escollit, AntDesign.
2. **Gestio de cookies:** Investigació i SetUp del sistema de cookies que s'utilitzarà per al sistema (universal-cookie). Primer es va utilitzar react-cookies però no funcionava correctament així que es va decidir canviar a universal-cookie.

Tasques de Back-End

Per la realització d'aquestes tasques s'ha seguit un tutorial d'AWS [22], en veure aquest tutorial es va decidir muntar el sistema amb l'arquitectura d'Amazon i s'ha seguit per tenir les nocions bàsiques quant als sistemes a utilitzar.

1. **Cognito configuration:** Investigació del funcionament i configuració de Cognito. Per tal de realitzar aquesta tasca s'han seguit diversos tutorials, cal destacar que en el Sprint següent, en crear les funcions de crear usuaris de manera externa (des de codi) es van fer certes modificacions.
2. **API Gateway configuration:** Investigació del funcionament i configuració de l'API Gateway. Per la realització d'aquesta tasca es va crear una API Gateway de prova per tal d'entendre el funcionament d'aquesta, no és fins al funcionament d'algunes lambdes (a l'Sprint 3) que es configura l'API Gateway definitiva.

3. **DynamoDB configuration:** Investigació del funcionament i configuració de DynamoDB. Igual que per l'API Gateway, és aquesta tasca es van seguir documentals per tal de fer proves i entendre el funcionament bàsic de DynamoDB, no és fins a l'Sprint 3 que es creen les primeres col·leccions.
4. **Lambdas configuration:** Investigació del funcionament i configuració de les diferents Lambdas. Igual que les tasques anteriors, s'han generat lambdes de prova i s'ha investigat com fer-ho amb golang, ja que aquest llenguatge no disposa d'un editor de text dins la consola d'AWS, sinó que s'han anat pujant els zips amb el codi per cada funció, no és fins a l'Sprint 3 que no es creen les primeres funcions lambda.
5. **S3 configuration:** Investigació del funcionament i configuració de S3. Durant aquesta tasca s'ha investigat el funcionament de l'emmagatzematge d'S3, en veure que és un sistema d'emmagatzematge de webs estàtiques i que s'havien d'importar els fitxers directament es va fer una cerca per tal de fer el desplegament de l'app de React, és per aquest motiu que es va afegir la tasca següent que s'encarrega d'això.
6. **Travis deploy configuration:** Configuració del desplegament amb Travis per tal que estigui hostejat a S3. En aquesta tasca es va configurar Travis per tal que el des dels commits de GitHub es faci el desplegament automàticament al sistema d'emmagatzematge d'S3 d'Amazon. [23]

Resum de les hores invertides al desenvolupament Back-End durant l'Sprint

Tasca	Hores previstes	Hores reals
Cognito configuration	3	4
API Gateway configuration	3	2
DynamoDB configuration	3	1
S3 configuration	3	3
Lambdas configuration	3	1
Travis deploy configuration	3	3
Hores fora de tasca		1
TOTAL		15

Taula 14.1: Hores invertides al desenvolupament Back-End durant l'Sprint 1

Resum de les hores invertides al desenvolupament Front-End durant l'Sprint

Tasca	Hores previstes	Hores reals
React SetUp	2	1
Gestio de cookies	2	3
Hores fora de tasca		5
TOTAL		9

Taula 14.2: Hores invertides al desenvolupament Front-End durant l'Sprint 1

TFG - COINK BACKLOG

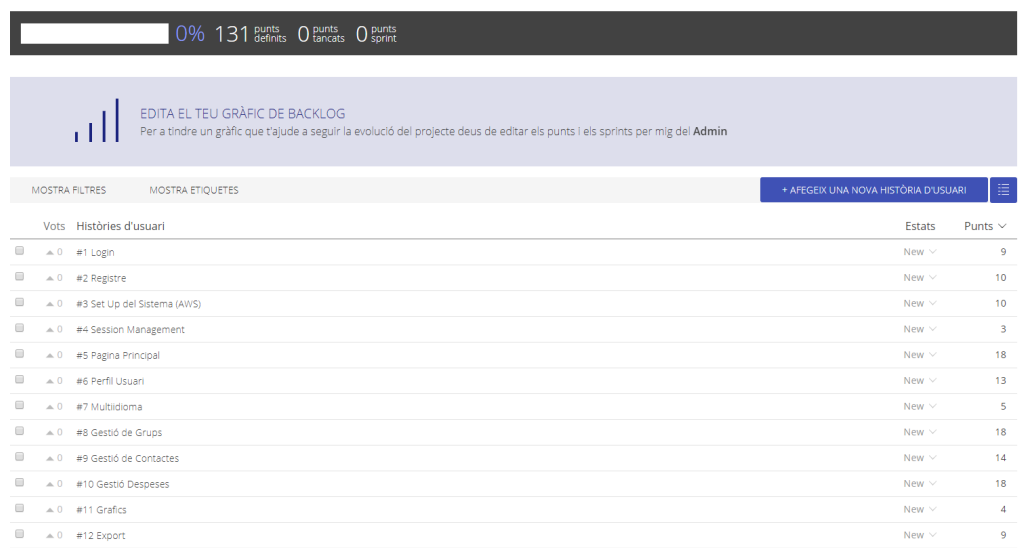


Figura 14.1: Backlog inicial - abans del Sprint 1

TFG - COINK SPRINT 1 04 MARÇ 2019-11 MARÇ 2019

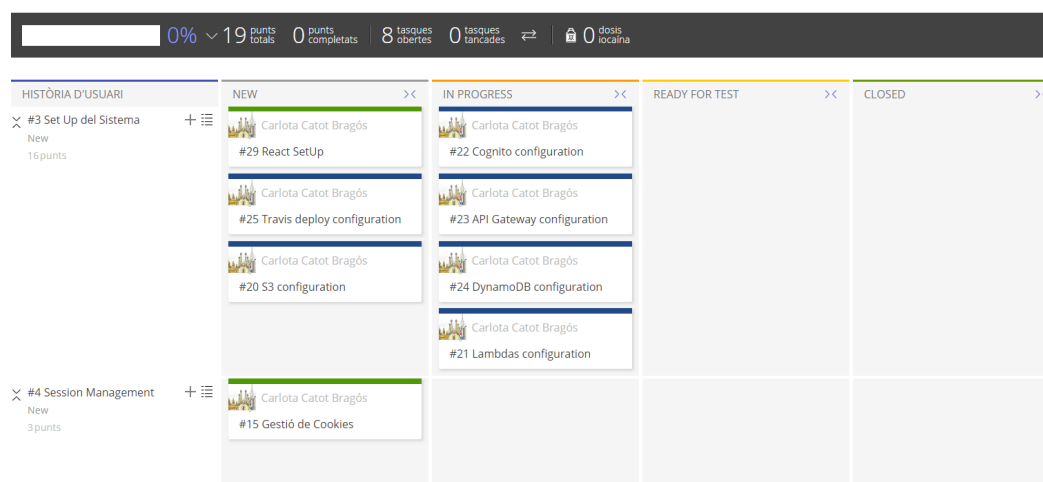


Figura 14.2: Panell de tasques de l'sprint 1

14.2 Sprint 2

Donat el backlog després del primer sprint (imatge 14.3) on respecte a l'sprint anterior i arran de la planificació realitzada a GEP s'han afegit noves històries d'usuari i s'han redefinit els punt de les històries, pel segon sprint s'ha decidit assignar les històries d'usuari **Login i Logout - 12 punts i Registre - 12 punts** i una durada de dues setmanes (del 11 de març del 2019 al 25 de març del 2019), les tasques assignades a aquest sprint (figura 14.4) consten de la part de desenvolupament tant de Front-End com de Back-End de tot el sistema dels usuaris. Durant aquest Sprint també s'estan realitzant paral·lelament les últimes tasques de documentació durant l'assignatura de GEP. Tota la informació de l'Sprint es pot consultar al taiga [24]

Tasques de Front-End

1. **Pantalla de login i logout:** Creació de la pantalla de login, on els usuaris entraran al sistema. Per tal de realitzar aquesta pantalla s'han utilitzat els components d'Ant-Design [25] d'un formulari de Login estàndard i s'ha modificat per tal que quedi maco, en la realització d'aquesta tasca també es va dissenyar el logo de l'aplicació. També s'ha creat un botó a la pantalla principal amb el menú superior i el botó de logout per tal de testejar la funcionalitat de Logout.
2. **Logica del registre:** Lògica del registre per tal que l'usuari es registri correctament, en aquesta tasca s'han dissenyat i desenvolupat les diferents pantalles necessàries això. Durant aquesta tasca s'han dissenyat i desenvolupat 2 de les pantalles de registre, la inicial on l'usuari entra les seves credencials i la de validació on l'usuari, un cop validat l'usuari s'accedirà a la pantalla principal del sistema. Per tal de crear les dues pantalles s'han seguit els requisits i criteris d'acceptació explicats anteriorment.

Tasques de Back-End

Pel desenvolupament de les funcions que es realitzaran durant aquestes tasques, s'ha utilitzat l'SDK de JavaScript per Amazon Cognito [26]. Totes aquestes funcions s'han creat a la classe cognito-config.js dins l'estructura de react (a l'Sprint 4 es migra a lambdas).

1. **Funció cognito login:** Funció encarregada de fer el login al sistema, al fer aquesta petició es torna un token d'identificació que per la part del Front-End s'emmagatzema a les cookies. Es comproven les credencials mitjançant peticions a cognito, per tant no cal fer la gestió de l'encryptació ni res per l'estil, per la part del Front-End s'esborren totes les dades de les cookies.
2. **Funció cognito logout:** Funció encarregada de fer logout al sistema, aquesta petició tancarà la sessió oberta de cognito.
3. **Funció cognito registre:** Funció encarregada de fer el registre del sistema
4. **Funció cognito account verification:** Funció encarregada de verificar el correu electrònic especificat al registre en el punt anterior, el codi es rep al correu electrònic, d'aquesta manera cognito verifica les credencials per poder a utilitzar el compte.
5. **Funció cognito reenviar codi de verificació:** Funció encarregada de reenviar el codi de verificació a l'e-mail especificat al registre en cas que l'usuari ho demani.

Resum de les hores invertides al desenvolupament Back-End durant l'Sprint

Tasca	Hores previstes	Hores reals
Funció cognito login	3	4
Funció cognito logout	1	2
Funció cognito account verification	2	2
Funció cognito reenviar codi de verificació	1	2
Funció cognito registre	3	2
Hores fora de tasca		1
TOTAL		13

Taula 14.3: Hores invertides al desenvolupament Back-End durant l'Sprint 2

Resum de les hores invertides al desenvolupament Front-End durant l'Sprint

Tasca	Hores previstes	Hores reals
Pantalla de login	4	4
Logica del registre	5	4
Hores fora de tasca		5
TOTAL		13

Taula 14.4: Hores invertides al desenvolupament Front-End durant l'Sprint 2

TFG - COINK BACKLOG

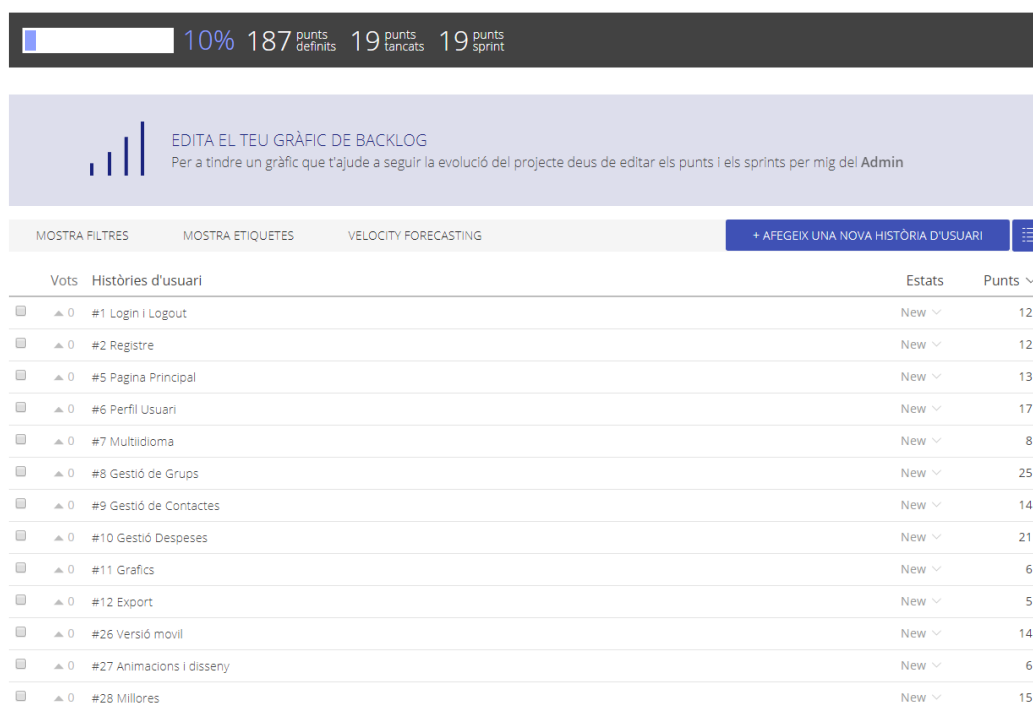


Figura 14.3: Backlog després de l'sprint 1

TFG - COINK SPRINT 2 11 MARÇ 2019-25 MARÇ 2019

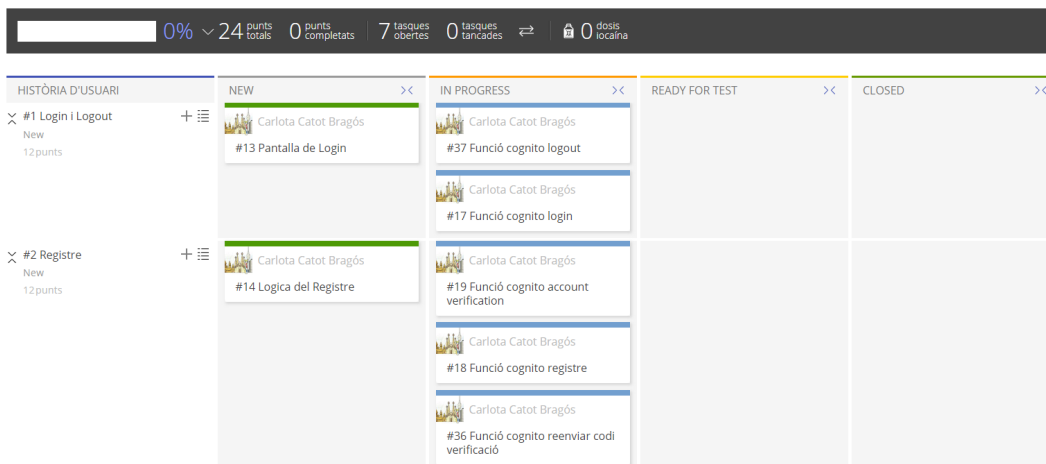


Figura 14.4: Panell de tasques de l'sprint 2

14.3 Sprint 3

Donat el backlog després del segon sprint (imatge 14.5), pel tercer sprint s'ha decidit assignar les històries d'usuari de **Gestió Despeses - 21 punts** i **Gràfics - 6 punts** i una durada de dues setmanes (del 25 de març del 2019 al 08 d'abril del 2019), les tasques assignades a aquest sprint (figura 14.6) consten de la part de desenvolupament tant de Front-End com de Back-End de tot el sistema de despeses, i també les gràfiques per mostrar de manera més gràfica les dades. Durant aquest sprint també es realitzaran tasques fora de les històries d'usuari assignades per tal de fer configuracions d'AWS. A l'inici d'aquest Sprint es realitzarà la presentació final de GEP, en concret el dia 27 de març. Tota la informació de l'Sprint es pot consultar al taiga [27].

Tasques de Front-End

1. **Lògica Activity:** Disseny de la lògica de tot el sistema de transaccions personals. Per tal de realitzar aquesta tasca es va pensar molt de quina manera seria usable, i seguint els requisits especificats a l'inici del desenvolupament, tot el sistema de transaccions. Es van realitzar nombrosos esborranys de les diferents pantalles i les diferents funcionalitats que aquestes havien de tenir.
2. **Pantalla all transactions:** Creació de la pantalla de totes les transaccions de l'usuari. Per tal de realitzar aquesta pantalla s'han utilitzat els components d'AntDesign [25] per crear una taula on apareixen totes les transaccions i s'han integrat les funcionalitats de cercar i ordenar per qualsevol dels camps especificats en els requisits del projecte.
3. **Pantalla year transactions:** Creació de la pantalla de transaccions per any. Per tal de realitzar aquesta pantalla s'han utilitzat els components d'AntDesign [25] per tal de crear dues taules (una per entrades i una per sortides) i les estadístiques de la suma total de cada taula per tal d'indicar a l'usuari el total d'entrades i de sortides realitzades en l'any. Igual que a la tasca anterior, s'han integrat les funcionalitats de cercar i ordenar per qualsevol dels camps especificats en els requisits del projecte.
4. **Editar transaction:** Creació de la funcionalitat que permet editar una transacció. Per tal de realitzar aquesta funcionalitat s'ha editat la taula creada a la tasca de *Pantalla all transactions* per tal fer que les siguis files siguin editables, s'ha afegit un botó que permet editar les files per tal que en clicar-lo, les files es tornen inputs on es pot editar la informació.
5. **Esborrar transaction:** Creació de la funcionalitat que permet esborrar una transacció. Per tal de realitzar aquesta funcionalitat s'ha editat la taula creada la tasca de *Pantalla all transactions* i s'ha afegit per cada fila un botó d'esborrar, en clicar-lo, apareix un missatge dient a l'usuari si estar segur d'esborrar la transacció, en cas afirmatiu s'esborra.
6. **Afegir transaction:** Creació de la funcionalitat que permet afegir transaccions. Per tal de realitzar aquesta funcionalitat s'han creat 2 botons dins la pantalla de totes les transaccions, un per afegir una entrada, i l'altre per afegir una sortida, en clicar-lo apareix un Modal per tal d'introduir les dades de la transacció que en ser creada s'afegiran a la taula.
7. **Lògica gràfiques:** Cerca d'una llibreria per mostrar les gràfiques i integració d'aquestes dins el sistema. Per tal de realitzar aquesta tasca es van fer proves amb moltes llibreries, però finalment es va optar per utilitzar la llibreria de ReCharts [28] per tal de dissenyar gràfiques de barres que mostrin la quantitat gastada per cada categoria. Aquestes gràfiques s'han implementat dins la pantalla de transaccions per any.

Tasques de Back-End

Per la realització de les tasques relacionades amb les lambdes, per cada lambda s'ha hagut de crear una funció, per tal de poder tenir la informació de l'API gateway a la lambda s'ha utilitzat un sistema de proxy [31], per tal de fer les operacions amb les dades de dynamoDB s'ha utilitzat el SDK de GO [30], per últim, com per la consola de les funcions lambda d'AWS no es permet crear el codi en GoLang en l'editor online, s'ha hagut de crear un fitxer zip per cadascuna de les funcions, per tal de crear aquests fitxers zip i que fossin acceptats per la consola de lambda s'ha hagut d'instal·lar el modul *build-lambda-zip* [29] i la manera de crear-los ha estat la següent:

```
> set GOOS=linux
> go build
> %GOPATH%/bin/build-lambda-zip -o main.zip executableName
```

** La primera comanda que canvia el sistema operatiu per generar el fitxer executable només és necessària en iniciar cmd*

1. **Definir endpoints:** Dissenyar els endpoints necessaris per a tot el sistema de transaccions personals. Per tal de realitzar aquesta tasca, s'han programat des de Swagger [32] els diferents endpoints necessaris per complir els requisits, aquests es poden consultar més en detall a la versió de Swagger 1.0.1 [33] (imatge 14.7).
2. **Configuracions definitives del sistema d'AWS:** Configuració d'API Gateway, les funcions de les lambdes (durant cada Sprint s'hauran de crear funcions noves) i les taules de DynamoDB (durant els següents Sprints caldrà afegir les taules necessàries). Després de l'Sprint 7, un cop acabat tot el desenvolupament es presenten en imatges com han quedat les diferents configuracions.
3. **Lambda all transactions:** Creació de l'endpoint *GET /transactions* i la funció lambda corresponent *getTransaction*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal d'obtenir les transaccions que pertanyen a un usuari concret. El nom de l'usuari s'agafa com a query parameter. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
4. **Lambda afegir transacció:** Creació de l'endpoint *POST /transactions* i la funció lambda corresponent *postTransaction*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal d'afegir noves dades en ella. Les dades a afegir és passen al body de la funció, mentre que el nom de l'usuari que s'ha de relacionar amb la transacció s'agafa com a query parameter. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
5. **Lambda editar transacció:** Creació de l'endpoint *PUT /transactions/{transactionID}* i la funció lambda corresponent *putTransaction*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal de modificar les dades. Les noves dades de la transacció s'agafen del body de la funció, mentre que les dades que identifiquen una transacció, que són l'identificador de la transacció i l'identificador d'usuari, s'agafen del path i d'un query parameter, respectivament. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
6. **Lambda esborrar transacció:** Creació de l'endpoint *DELETE /transactions/{transactionID}* i la funció lambda corresponent *deleteTransaction*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal d'esborrar les dades. Les dades que identifiquen una transacció són l'identificador de la transacció i l'identificador d'usuari, aquestes s'agafen del path i d'un query parameter, respectivament. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.

7. **Lambda transaccions per anys:** Creació de l'endpoint *GET /transactions/byYear/{year}* i la funció lambda corresponent *getYearTransaction*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal d'obtenir les transaccions que pertanyen a un usuari concret en un any concret. El nom de l'usuari s'agafa com a query parameter, mentre que l'any a cercar s'agafa com a path parameter. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
8. **Carregar dades des de diferents lambdas:** Actualització de l'endpoint de *GET /transactions/byYear/{year}* per tal d'obtenir les dades necessàries per carregar les gràfiques. Per tal d'implementar aquesta funció s'ha modificat el JSON de retorn de l'endpoint per tal de tornar per cada categoria el que s'ha gastat o ha guanyat depenent del tipus de transacció. Aquesta modificació està reflectida al Swagger amb el Model *TransactionYear*. Un modificat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
9. **CloudFront configuration:** Configuració del sistema CDN per accedir a la WebApp. Per tal de realitzar aquesta tasca s'ha configurat un sistema per així no entrar a la web mitjançant el sistema d'S3 [34].

Resum de les hores invertides al desenvolupament Back-End durant l'Sprint

Tasca	Hores previstes	Hores reals
Definir endpoints	7	8
Lambda all transactions	3	3
Lambda afegir transaccio	3	3
Lambda editar transacció	3	3
Lambda esborrar transaccio	3	3
CloudFront configuration	4	5
Lambda transaccio per anys	5	4
Configuracions definitives del sistema d'AWS	5	6
Carregar dades desde diferents lambdas	3	2
Hores fora de tasca		2
TOTAL		39

Taula 14.5: Hores invertides al desenvolupament Back-End durant l'Sprint 3

Resum de les hores invertides al desenvolupament Front-End durant l'Sprint

Tasca	Hores previstes	Hores reals
Logica Activity	15	18
Pantalla All transactions	7	9
Pantalla year transactions	7	8
Editar transaction	3	4
Esborrar transaction	2	2
Afegir transaction	4	5
Logica grafiques	4	6
Hores fora de tasca		5
TOTAL		57

Taula 14.6: Hores invertides al desenvolupament Front-End durant l'Sprint 3

TFG - COINK BACKLOG

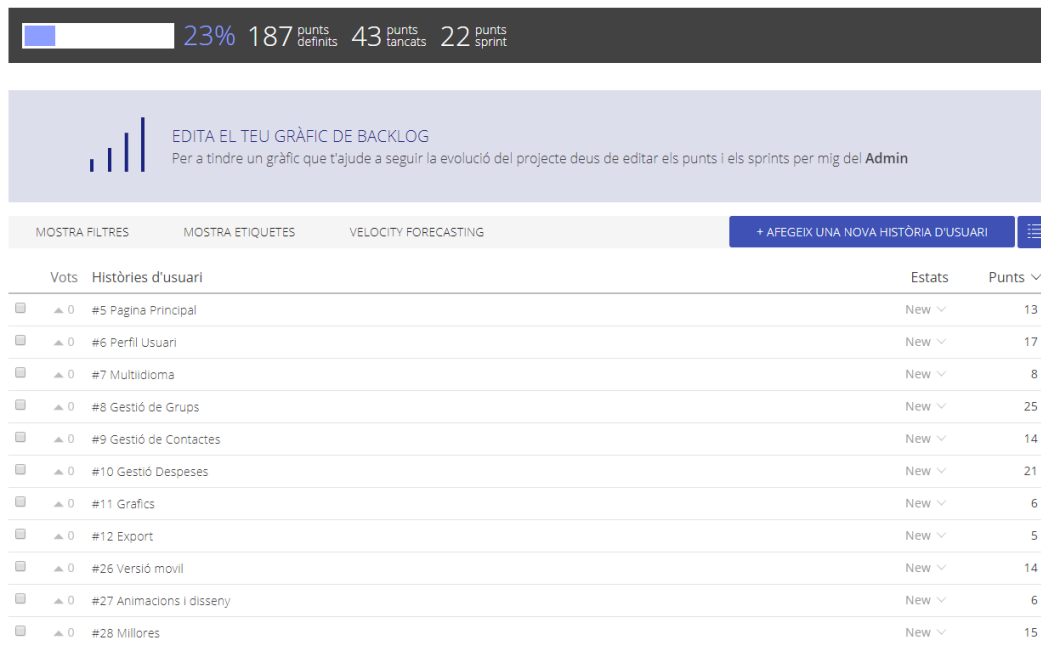


Figura 14.5: Backlog després de l'sprint 2

TFG - COINK SPRINT 3 25 MARÇ 2019-08 ABR. 2019

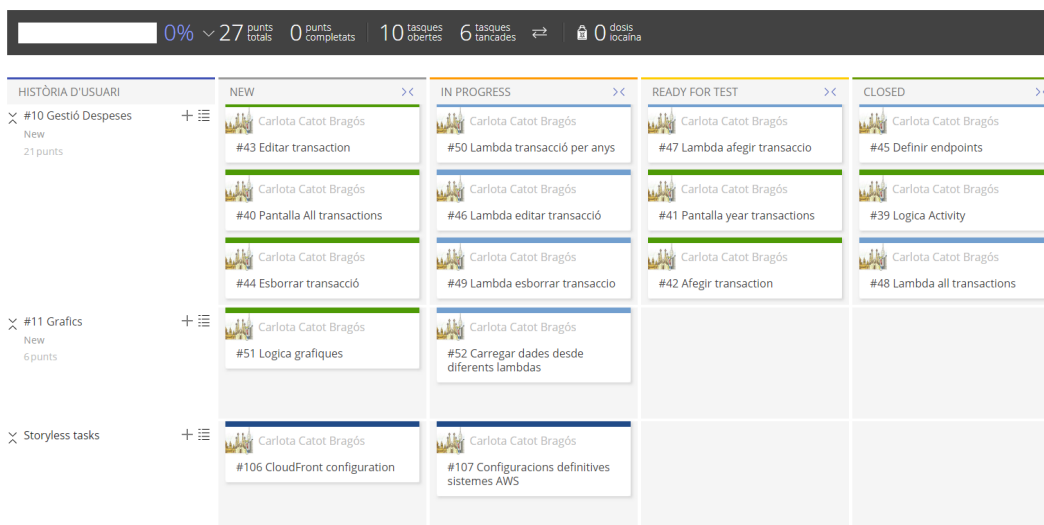


Figura 14.6: Panell de tasques de l'sprint 3

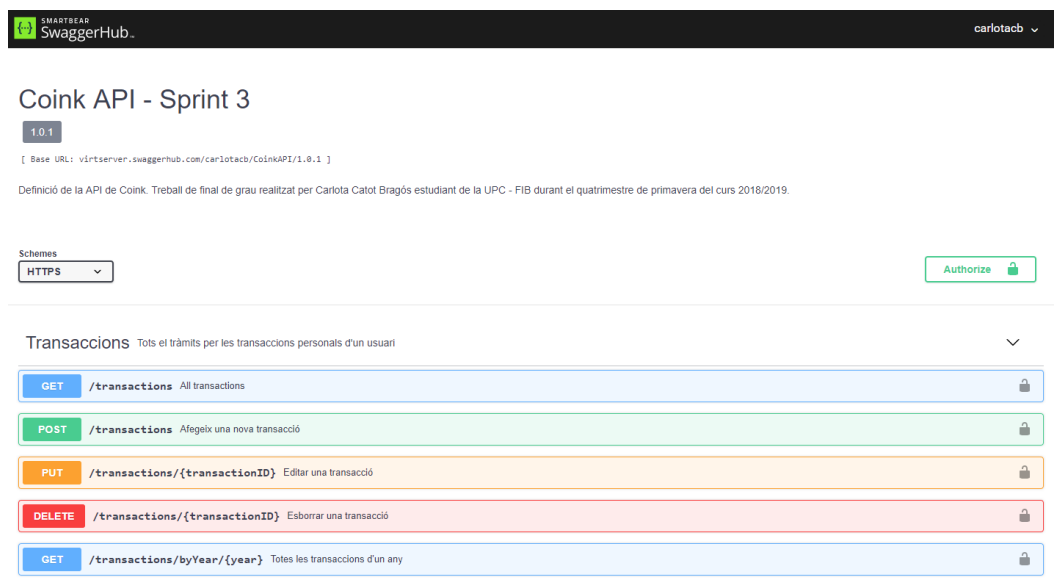


Figura 14.7: Endpoints dissenyats per l'Sprint 3

14.4 Sprint 4

Donat el backlog després del tercer sprint (imatge 14.8), pel quart sprint s'ha decidit assignar les històries d'usuari de **Perfil Usuari - 17 punts** i **Pàgina Principal - 13 punts** i una durada de dues setmanes (del 08 d'abril del 2019 al 22 d'abril del 2019), les tasques assignades a aquest sprint (figura 14.9) consten de la part de desenvolupament tant de Front-End com de Back-End de tot el sistema de consulta i modificació de les dades de l'usuari, i també la pantalla principal del sistema. Durant aquest sprint també es realitzaran tasques fora de les històries d'usuari a causa de problemes de seguretat en el sistema de funcions relacionades amb cognito. Tota la informació de l'Sprint es pot consultar al taiga [35]

Tasques de Front-End

1. **Pantalla informació usuari:** Creació de la pantalla d'informació de l'usuari. Per realitzar aquesta tasca, en primer lloc es va pensar de quina manera seria usable per l'usuari gestiona la informació del seu perfil, i és per això que, utilitzant els sistemes d'AntDesign [25] s'ha creat una pantalla amb un formulari on a l'inici apareixen les dades actuals de l'usuari, amb els camps modificables disponibles per fer canvis i un botó per tal de guardar les dades.
2. **Canviar password usuari:** Creació de la funcionalitat de canviar la contrasenya de l'usuari. Per realitzar aquesta tasca s'ha afegit un botó a la pantalla d'informació de l'usuari que en clicar-lo apareix un modal, introduint les dades demanades (contrasenya actual i nova contrasenya) es canvia la contrasenya de l'usuari que per tant, podrà utilitzar la nova contrasenya als futurs inicis de sessió.
3. **Pantalla principal:** Creació de la pantalla principal del sistema. Per tal de realitzar aquesta tasca, en primer lloc es va pensar que necessita un usuari a la pantalla principal. Un cop realitzats diversos esborranys i decidides les dades a introduir s'ha desenvolupat la pantalla. A la part superior de la pantalla s'ha decidit afegir una frase per tal de motivar a l'usuari a estalviar [36], pel que fa a la resta de camps, s'han posat estadístiques dels grups (que per al moment no tenen dades) i de les transaccions.
4. **Estadístiques de l'any:** Integració de gràfiques a la pantalla principal. Per tal de realitzar aquesta tasca s'ha reaprofitat codi de l'Sprint anterior utilitzant la llibreria de ReCharts [28] per tal de mostrar les mateixes gràfiques que es mostren a la pantalla d'activitat, en aquest cas per l'any actual.

5. **Pantalla iniciar usuari al sistema:** Creació de la pantalla per inicialitzar les dades de l'usuari. En realitzar la pantalla principal del sistema, es va pensar que l'usuari necessitaria introduir el saldo que té en aquell moment per tal d'inicialitzar el sistema, és per això que es va crear una pantalla extra al procés de registre. Per tal de saber quan és el primer cop que l'usuari entra al sistema també s'ha creat la 3a pantalla del procés on es fa un fals login.

Tasques de Back-End

Per la realització de les tasques relacionades amb les lambdes, s'han utilitzat els mateixos recursos que als Sprints anteriors. Pel que fa a les funcions migrades, s'han mantingut en JavaScript, i també s'ha programat amb aquest llenguatge totes les funcions relacionades amb els usuaris (obtenir i editar informació de l'usuari i canviar la contrasenya) en aquest Sprint per tal de seguir el mateix patró. El codi programat en JavaScript és editable a la consola online d'AWS per tant no cal realitzar cap procés per pujar el codi mitjançant un fitxer zip. Pel que fa a la configuració d'AWS per aquest sprint s'ha creat la taula de Users i de Balance a DynamoDB.

1. **Definir endpoints:** Dissenyar els endpoints necessaris per a tot el sistema de transaccions personals. Per la realització d'aquesta tasca, s'han programat des de Swagger [32] els diferents endpoints necessaris per complir els requisits, aquests es poden consultar més en detall a la versió de Swagger 1.0.2 [37] (imatge 14.10).
2. **Lambda informació usuari:** Creació de l'endpoint *GET /users/updateInformation+/{username}* i la funció lambda corresponent *getUpdateUserInformation*. Per tal d'implementar aquesta funció s'ha implementat un sistema per accedir a les dades de l'usuari a Cognito. El nom de l'usuari s'agafa com a query parameter. Com la funció ha estat implementada en JavaScript no és necessari generar cap zip de manera especial, s'ha implementat des de la consola d'AWS.
3. **Lambda editar informació usuari:** Creació de l'endpoint *PUT /users/updateInformation/{username}* i la funció lambda corresponent *putUpdateInformation*. Per tal d'implementar aquesta funció s'ha implementat un sistema per poder canviar les dades de l'usuari tant a Cognito com a la base de dades dels usuaris. El nom de l'usuari s'agafa com a query parameter, mentre que les noves dades de l'usuari es passen com a body de la funció. Com la funció ha estat implementada en JavaScript no és necessari generar cap zip de manera especial, s'ha implementat des de la consola d'AWS.
4. **Lambda canviar password:** Creació de l'endpoint *PUT /users/changePassword+/{username}* i la funció lambda corresponent *changePasswordLambda*. Per tal d'implementar aquesta funció s'ha implementat un sistema per poder canviar les dades d'inici de sessió de l'usuari a Cognito. El nom de l'usuari s'agafa com a query parameter, mentre que les dades necessàries per canviar la contrasenya es passen com a body de la funció. Com la funció ha estat implementada en JavaScript no és necessari generar cap zip de manera especial, s'ha implementat des de la consola d'AWS.
5. **Lambda balanç:** Creació de l'endpoint *GET /balance* i la funció lambda corresponent *getBalance*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal d'obtenir el balanç corresponent a l'usuari actual. El nom de l'usuari s'agafa com a query parameter. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
6. **Lambda inicialitzar usuari:** Creació de l'endpoint *POST /users/init* i la funció lambda corresponent *postBalance*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal d'afegir noves dades en ella. Les dades a afegir es passen al body de la funció, mentre que el nom de l'usuari que s'ha de relacionar amb el balance s'agafa com a query parameter. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
7. **Migrar tot el sistema de conigito-config.js a Lambdas:** Canviar totes les funcions del Sprint 2 a lambdes. Durant aquesta tasca s'han creat diversos endpoints relacionats amb les funcions del procés de login i registre.

- (a) Per la funció cognito login s'ha creat l'endpoint *POST /users/login* amb la funció lambda *loginLambda*.
- (b) Per la funció cognito logout s'ha creat l'endpoint *POST /users/logout* amb la funció lambda *logoutLambda*.
- (c) Per la funció cognito registre s'ha creat l'endpoint *POST /users/register* amb la funció lambda *registerLambda*.
- (d) Per la funció account verification s'ha creat l'endpoint *POST /users/confirm* amb la funció lambda *confirmRegistrationLambda*.
- (e) Per la funció account verification s'ha creat l'endpoint *POST /users/confirm/resend* amb la funció lambda *resendCodeLambda*.

Per la creació de tots els endpoints s'ha intentat mantenir al màxim la configuració original tot i que s'ha hagut de fer modificacions per tal de fer-les funcions lambda.

Resum de les hores invertides al desenvolupament Back-End durant l'Sprint

Tasca	Hores previstes	Hores reals
Definir endpoints	5	5
Lambda informació usuari	4	3
Lambda editar informació usuari	4	5
Lambda inicialitzar usuari	3	3
Migrar tot el sistema	6	7
Lambda canviar password	4	3
Lambda balanç	3	2
Hores fora de tasca		5
TOTAL		33

Taula 14.7: Hores invertides al desenvolupament Back-End durant l'Sprint 4

Resum de les hores invertides al desenvolupament Front-End durant l'Sprint

Tasca	Hores previstes	Hores reals
Pantalla informació usuari	5	6
Canviar password usuari	2	2
Estadístiques de l'any	2	1
Pantalla principal	2	3
Pantalla inciar usuari al sistema	3	4
Hores fora de tasca		5
TOTAL		21

Taula 14.8: Hores invertides al desenvolupament Front-End durant l'Sprint 4

TFG - COINK BACKLOG

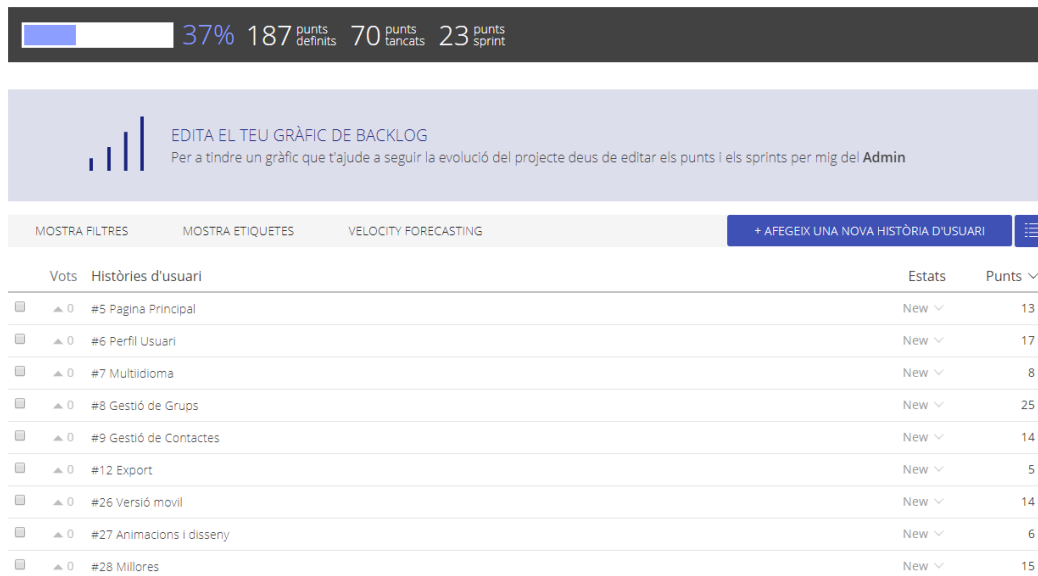


Figura 14.8: Backlog després de l'sprint 3

TFG - COINK SPRINT 4 08 ABR. 2019-22 ABR. 2019

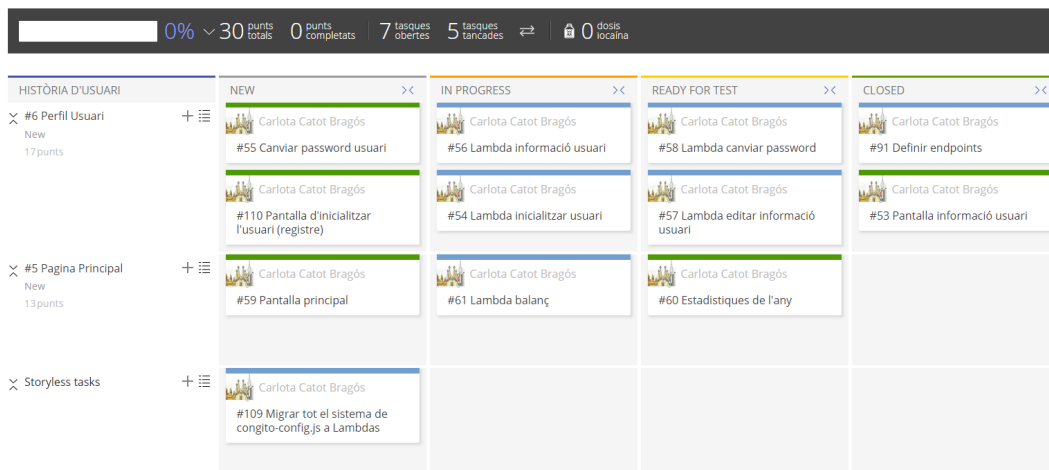


Figura 14.9: Panell de tasques de l'sprint 4

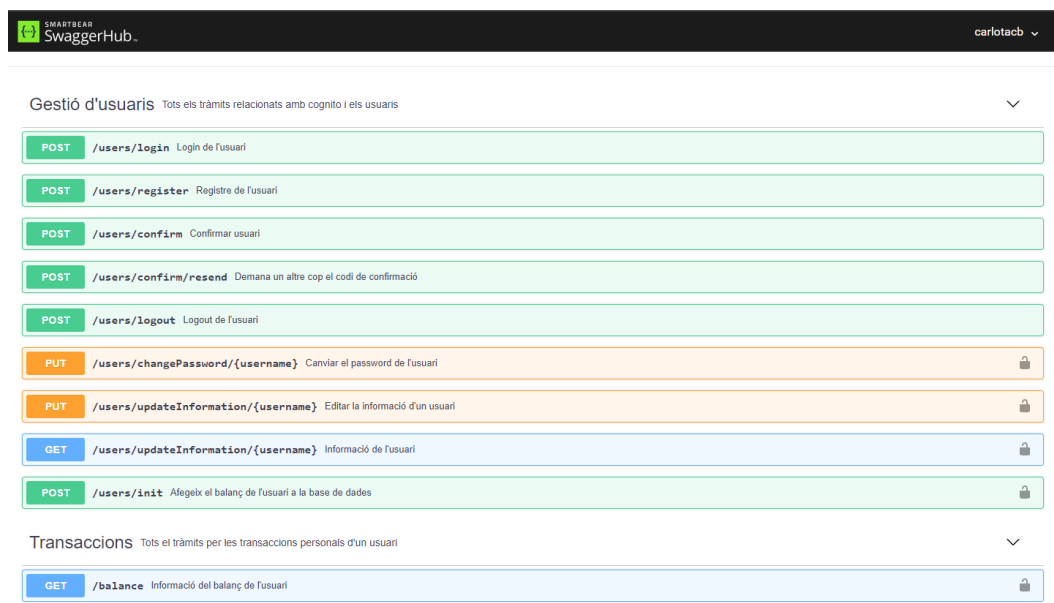


Figura 14.10: Endpoints dissenyats per l'Sprint 4

14.5 Sprint 5

Donat el backlog després del quart sprint (imatge 14.11), pel quart sprint s'ha decidit assignar les històries d'usuari de **Gestió de contactes - 14 punts** i **Gestió de grups - 25 punts** i una durada de dues setmanes (del 22 d'abril del 2019 al 06 de maig del 2019), les tasques assignades a aquest sprint (figura 14.12) consten de la part de desenvolupament tant de Front-End com de Back-End de tot el sistema de despeses compartides, tant amb la part de contactes com amb la part de grups. Tota la informació de l'Sprint es pot consultar al taiga [38]

Tasques de Front-End

1. **Pantalla contactes:** Creació de la pantalla de contactes. Per realitzar aquesta tasca, en primer lloc es va pensar de quina manera podríem mostrar els usuaris a la pantalla i quines funcionalitats es podien oferir. Utilitzant els sistemes d'AntDesign [25] s'ha creat una pantalla amb una llista on es poden veure tots els contactes i les gestions pendents amb cadascun. També es pot eliminar o afegir un contacte.
2. **Afegir un contacte:** Creació de la funcionalitat d'afegir un contacte. Per realitzar aquesta tasca s'ha afegit un botó a la part superior de la *pantalla de contactes* que en clicar-lo apareix un modal. En aquest modal apareix un camp autocompletable que permet veure tots els usuaris que hi ha al sistema per tal d'afegir un usuari vàlid.
3. **Esborrar un contacte:** Creació de la funcionalitat d'eliminar un contacte. Per tal de realitzar aquesta funcionalitat s'ha editat la llista creada la tasca de *Pantalla contactes* i s'ha afegit per cada element de la llista un botó d'esborrar, en clicar-lo, apareix un missatge dient a l'usuari si estar segur d'esborrar el contacte, en cas afirmatiu s'esborra.
4. **Logica grups:** Disseny de la lògica de tot el sistema de grups. Per tal de realitzar aquesta tasca es va pensar molt de quina manera seria usable, i seguint els requisits especificats a l'inici del desenvolupament, tot el sistema de gestió de grups. Es van realitzar nombrosos esborranys de les diferents pantalles i les diferents funcionalitats que aquestes havien de tenir.
5. **Pantalla summary:** Creació de la pantalla de resum de grups. Per tal de realitzar aquesta pantalla s'han utilitzat els components d'AntDesign per tal de crear una llista desplegable amb els grups. Per cada element de la llista, si no està desplegat, es veu només el títol d'aquest, en desplegar l'element de la llista es poden realitzar múltiples

funcionalitats, esborrar el grup, veure més informació d'aquest, pagar als usuaris que es deu diners.

6. **Pantalla informació grup:** Creació de la pantalla d'informació d'un grup. Per tal de realitzar aquesta tasca s'han utilitzat els components d'AntDesign per crear una taula on apareixen tots els moviments d'un grup i s'han integrat les funcionalitats de cercar i ordenar per qualsevol dels camps especificats en els requisits del projecte.
7. **Editar informació grup:** Creació de la pantalla group information. Per realitzar aquesta tasca, s'han utilitzat els sistemes d'AntDesign per crear una pantalla molt semblant a la d'informació de l'usuari on apareixen les dades actuals del grup, amb els camps modificables disponibles per fer canvis i un botó per tal de guardar les dades.
8. **Crear grup:** Creació de la funcionalitat de crear un grup. Per realitzar aquesta tasca s'ha afegit un botó a la part superior de la *pantalla de grups* que en clicar-lo apareix un modal. En aquest modal apareixen diversos camps per tal d'introduir el títol i la descripció del grup i un botó per poder anar afegint els participants del grup (només estan disponibles els contactes de l'usuari).
9. **Afegir moviment:** Creació de la funcionalitat de crear un grup. Per realitzar aquesta tasca s'ha afegit un botó a la part superior de la *pantalla d'informació d'un grup* que en clicar-lo apareix un modal. En aquest modal apareixen diversos camps per tal d'introduir el títol, la data, el pagador, la descripció i la quantitat del moviment i un botó per poder anar afegint els participants del grup (només estan disponibles els usuaris que pertanyen al grup).
10. **Editar moviment:** Creació de la funcionalitat que permet editar un moviment. Per tal de realitzar aquesta funcionalitat s'ha editat la taula creada a la tasca de *pantalla d'informació d'un grup* per tal fer que les siguis files siguin editables, s'ha afegit un botó que permet editar les files per tal que en clicar-lo, les files es tornen inputs on es pot editar la informació.
11. **Pantalla participants grup:** Creació de la pantalla de participants d'un grup. Per tal de realitzar aquesta pantalla s'han utilitzat els components d'AntDesign per tal de crear una llista desplegable amb tots els participants del grup. Per cada element de la llista, si no està desplegat, es veu només el nom d'usuari del participant, en desplegar l'element de la llista es poden veure el que deu i el que li deuen a l'usuari desplegat.

Tasques de Back-End

Per la realització de les tasques relacionades amb les lambdes, s'han utilitzat els mateixos recursos que als l'Sprints anteriors.

1. **Definir endpoints:** Dissenyar els endpoints necessaris per a tot el sistema de contactes i de grups. Per la realització d'aquesta tasca, s'han programat des de Swagger [32] els diferents endpoints necessaris per complir els requisits, aquests es poden consultar més en detall a la versió de Swagger 1.0.3 [39] (imatge 14.13).
2. **Lambda tots els contactes:** Creació de l'endpoint `GET /contacts/{userID}` i la funció lambda corresponent `getContacts`. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal d'obtenir els contactes relacionats amb l'usuari actual. El nom de l'usuari s'agafa com a path parameter. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
3. **Lambda tots els usuaris:** Creació de l'endpoint `GET /users` i la funció lambda corresponent `getUsers`. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal d'obtenir tots els usuaris que pertanyen al sistema. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.

4. **Lambda afegir contacte:** Creació de l'endpoint *POST /contacts* i la funció lambda corresponent *postContact*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal d'afegir noves dades en ella. Les dades a afegir es passen al body de la funció, mentre que el nom de l'usuari que s'ha de relacionar amb el contacte s'agafa com a query parameter. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
5. **Lambda esborrar contacte:** Creació de l'endpoint *DELETE /contacts/{userID} + /{contactID}* i la funció lambda corresponent *deleteContact*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal d'esborrar les dades. Les dades que identifiquen un contacte són l'identificador de l'usuari i l'identificador del contacte, aquestes s'agafen dels path parameters. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
6. **Lambda crear grup:** Creació de l'endpoint *POST /groups* i la funció lambda corresponent *postGroup*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal d'afegir noves dades en ella. Les dades a afegir es passen al body de la funció. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
7. **Lambda editar grup:** Creació de l'endpoint *PUT /groups/details/{groupID}* i la funció lambda corresponent *putGroups*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal de modificar les dades. Les noves dades del grup s'agafen del body de la funció, mentre que l'identificador del grup s'agafa del path parameter. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
8. **Lambda esborrar grup:** Creació de l'endpoint *DELETE /groups/details/{groupID}* i la funció lambda corresponent *deleteGroup*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal d'esborrar les dades. L'identificador del grup s'agafa del path parameter. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
9. **Lambda grups d'un usuari:** Creació de l'endpoint *GET /groups/{username}* i la funció lambda corresponent *getGroup*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal d'obtenir les dades necessàries de tots els grups als quals pertany l'usuari. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
10. **Lambda informació del grup (dades + moviments):** Creació de l'endpoint *GET /groups/details/{groupID}* i la funció lambda corresponent *getGroupInformation*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal d'obtenir totes les dades d'un grup. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
11. **Lambda afegir moviment:** Creació de l'endpoint *POST /groups/details/{groupID}* i la funció lambda corresponent *postGroupMovement*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal d'afegir noves dades en ella. Les dades a afegir es passen al body de la funció, i l'identificador del grup amb el qual s'ha de relacionar el moviment és s'agafa del path parameter. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
12. **Lambda editar moviment:** Creació de l'endpoint *PUT /groups/details/{groupID} + /movements/{movementID}* i la funció lambda corresponent *putGroupMovement*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal de modificar les dades. Les noves dades del moviment s'agafen del body de la funció, mentre que els identificadors del grup i del moviment s'agafen dels path parameters. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.

13. **Lambda esborrar moviment:** Creació de l'endpoint *DELETE /groups/details+/{groupID}/movements/{movementID}* i la funció lambda corresponent *deleteGroupMovement*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal d'esborrar les dades. Els identificadors del grup i del moviment s'agafen del path parameter. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.
14. **Lambda saldar deutes:** Creació de l'endpoint *PUT /groups/details/{groupID}+ /movements/{movementID}/pay* i la funció lambda corresponent *putPayGroupMovement*. Per tal d'implementar aquesta funció s'han realitzat les peticions necessàries a la base de dades per tal de modificar les dades del pagament i afegir un nou moviment. Els identificadors del grup i del moviment s'agafen dels path parameters, ja que les dades a canviar només és el booleà que indica si l'usuari ha pagat. Un cop creat el codi s'ha pujat a la lambda corresponent per poder tenir accés des de l'endpoint.

Resum de les hores invertides al desenvolupament Back-End durant l'Sprint

Tasca	Hores previstes	Hores reals
Lambda tots els contactes	10	12
Lambda tots els usuaris	3	3
Lambda afegir contacte	4	3
Lambda esborrar contacte	3	3
Lambda crear grup	3	3
Lambda editar grup	3	3
Lambda esborrar grup	3	3
Lambda grups d'un usuari	5	4
Lambda informació del grup (dades + moviments)	5	7
Lambda afegir moviment	4	7
Lambda editar moviment	4	5
Lambda esborrar moviment	4	5
Lambda saldar deutes	3	7
Hores fora de tasca		7
TOTAL		75

Taula 14.9: Hores invertides al desenvolupament Back-End durant l'Sprint 5

Resum de les hores invertides al desenvolupament Front-End durant l'Sprint

Tasca	Hores previstes	Hores reals
Pantalla contactes	7	7
Afegir un contacte	2	2
Esborrar un contacte	2	1
Logica grups	15	22
Pantalla summary	7	8
Pantalla informació grup	6	8
Editar informació grup	4	4
Crear grup	4	4
Afegir moviment	4	4
Editar moviment	4	4
Pantalla participants grup	7	8
Hores fora de tasca		6
TOTAL		78

Taula 14.10: Hores invertides al desenvolupament Front-End durant l'Sprint 5

TFG - COINK BACKLOG

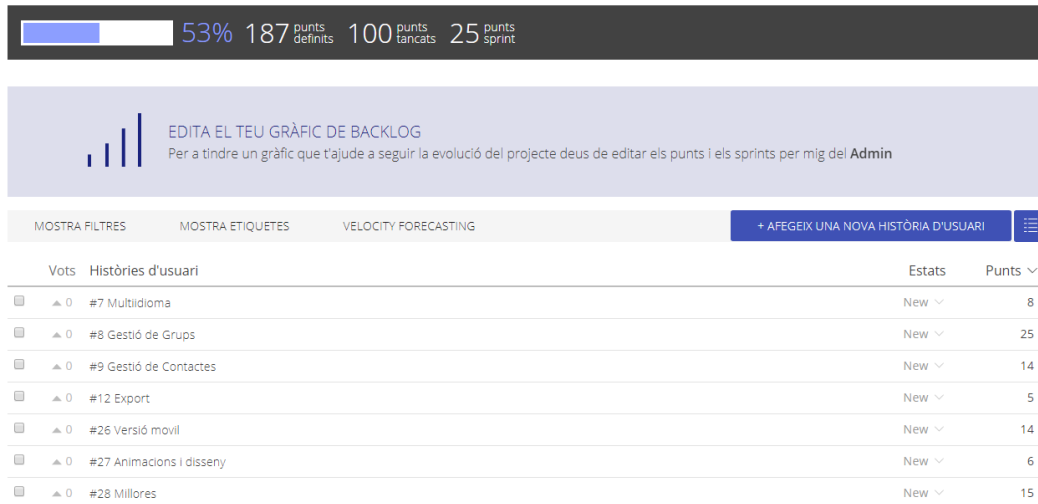


Figura 14.11: Backlog després de l'sprint 4

TFG - COINK SPRINT 5 22 ABR. 2019-06 MAIG 2019

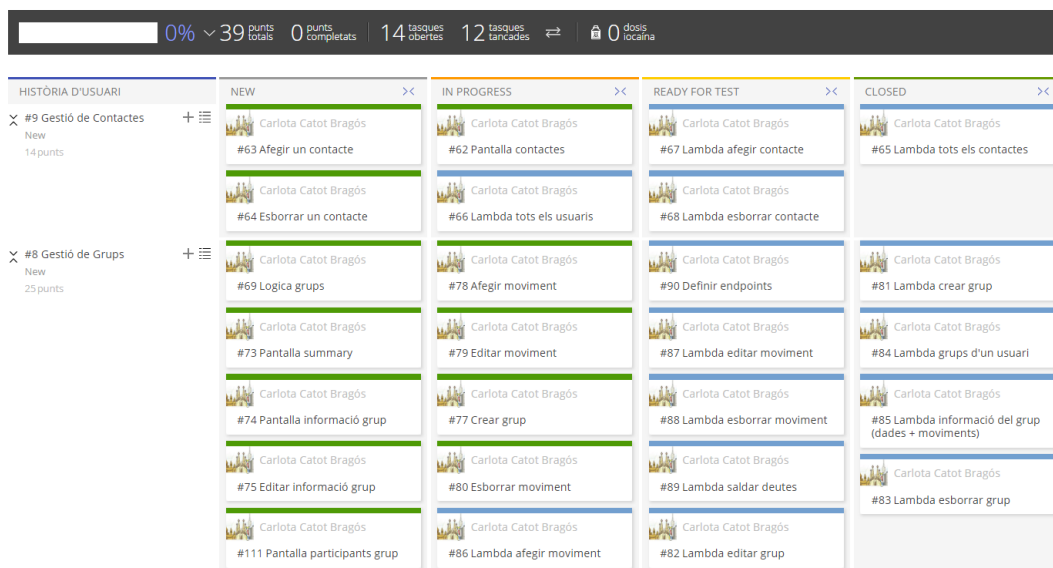


Figura 14.12: Panell de tasques de l'sprint 5

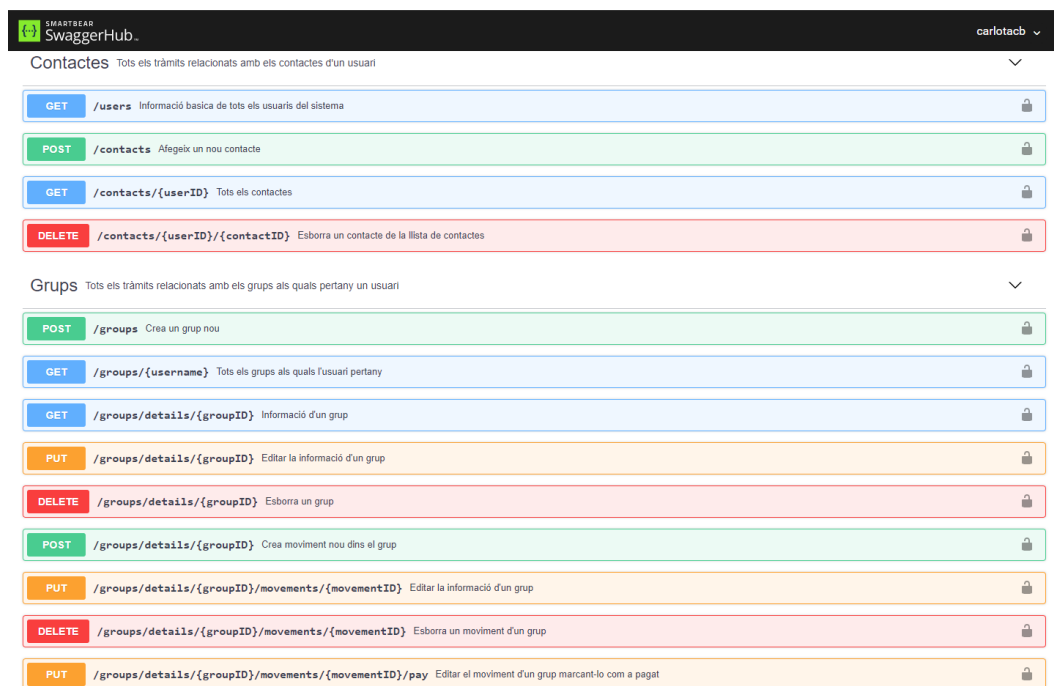


Figura 14.13: Endpoints dissenyats per l'Sprint 5

14.6 Sprint 6

Donat el backlog després del cinquè sprint (imatge 14.14), pel sisè sprint s'ha decidit assignar les històries d'usuari de **Export - 5 punts**, **Multiidioma - 8 punts** i **Versió mòbil - 14 punts** i una durada de tres setmanes (del 06 de maig del 2019 al 20 de maig del 2019), les tasques assignades a aquest sprint (figura 14.15) consten de la part de desenvolupament a la part de Front-End del desenvolupament de les pantalles per tal que l'aplicació compleixi l'objectiu de ser una webApp tenint la seva versió mòbil i funcionalitats extres com ens l'Export i el multiidioma. Per aquest sprint s'han assignat 3 setmanes, ja que pels Sprints anteriors no es van acabar les tasques a temps i s'han de completar de manera paral·lela. Tota la informació de l'Sprint es pot consultar al taiga [40].

Tasques de Front-End

Per la realització de totes les tasques del Front-End s'han utilitzat els sistemes d'AntDesign [25]

1. **Crear export per transaccions usuari:** Creació de la funcionalitat d'exportar les transaccions d'un usuari. Per realitzar aquesta funcionalitat s'ha modificat la pantalla de *totes les transaccions* per afegir un botó que, al clicar-lo es descarregarà un fitxer Excel equivalent a la taula de transaccions.
2. **Crear export pels moviments d'un grup:** Creació de la funcionalitat d'exportar les transaccions d'un usuari. Per realitzar aquesta funcionalitat s'ha modificat la pantalla de *informació d'un grup* per afegir un botó que, en clicar-lo es descarregarà un fitxer Excel equivalent a la taula de moviments d'un grup.
3. **Logica multiidioma (i18n):** Disseny del sistema multiidioma del sistema. Per realitzar aquesta tasca s'han investigat diverses llibreries que permeten el multidioma, finalment s'ha decidit integrar la llibreria d'i18n [42].
4. **Traduir tots els strings al català:** Traducció de la web al català. Per tal de realitzar aquesta tasca s'ha creat un fitxer amb tots els strings de les diferents pantalles en català.

5. **Traduir tots els strings al castellà:** Traducció de la web al castellà. Per tal de realitzar aquesta tasca s'ha creat un fitxer amb tots els strings de les diferents pantalles en castellà.
6. **Traduir tots els strings a l'anglès:** Traducció de la web a l'anglès. Per tal de realitzar aquesta tasca s'ha creat un fitxer amb tots els strings de les diferents pantalles en anglès.
7. **Pantalla login:** Creació de la versió web de la pantalla de login. Per tal de realitzar aquesta tasca s'ha creat una adaptació per mòbil del sistema web.
8. **Pantalla registre (tot el sistema):** Creació de la versió web de les pantalles de registre. Per tal de realitzar aquesta tasca s'ha creat una adaptació per mòbil del sistema web.
9. **Pantalla activity (múltiples pantalles):** Creació de la versió web de les pantalles d'activitat. Per tal de realitzar aquesta tasca s'ha creat una adaptació per mòbil del sistema web.
10. **Pantalla principal:** Creació de la versió web de la pantalla principal. Per tal de realitzar aquesta tasca s'ha creat una adaptació per mòbil del sistema web.
11. **Pantalla grups (múltiples pantalles):** Creació de la versió web de les pantalles de grups. Per tal de realitzar aquesta tasca s'ha creat una adaptació per mòbil del sistema web.
12. **Pantalla perfil:** Creació de la versió web de la pantalla de perfil. Per tal de realitzar aquesta tasca s'ha creat una adaptació per mòbil del sistema web.
13. **Pantalla contactes:** Creació de la versió web de la pantalla de contactes. Per tal de realitzar aquesta tasca s'ha creat una adaptació per mòbil del sistema web.

Resum de les hores invertides al desenvolupament Front-End durant l'Sprint

Tasca	Hores previstes	Hores reals
Crear export per transaccions usuari	4	6
Crear export per moviments d'un grup	1	1
Logica multiidioma (i18n)	7	8
Traduir al català	2	2
Traduir al castellà	3	3
Traduir a l'anglès	3	3
Pantalla login	4	1
Pantalla registre	7	2
Pantalla activity	10	11
Pantalla principal	4	1
Pantalla grups	10	12
Pantalla perfil	4	1
Pantalla contactes	4	2
Hores fora de tasca		6
TOTAL		59

Taula 14.11: Hores invertides al desenvolupament Front-End durant l'Sprint 6

TFG - COINK BACKLOG

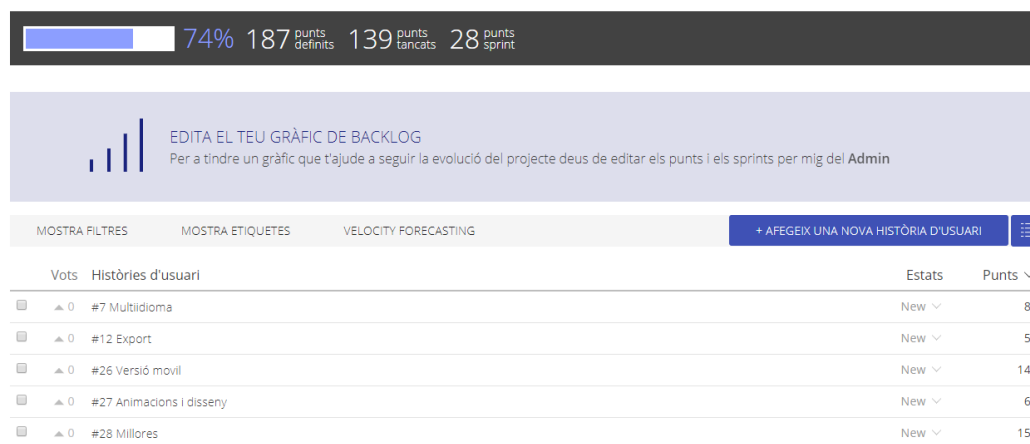


Figura 14.14: Backlog després de l'sprint 5

TFG - COINK SPRINT 6 06 MAIG 2019-20 MAIG 2019

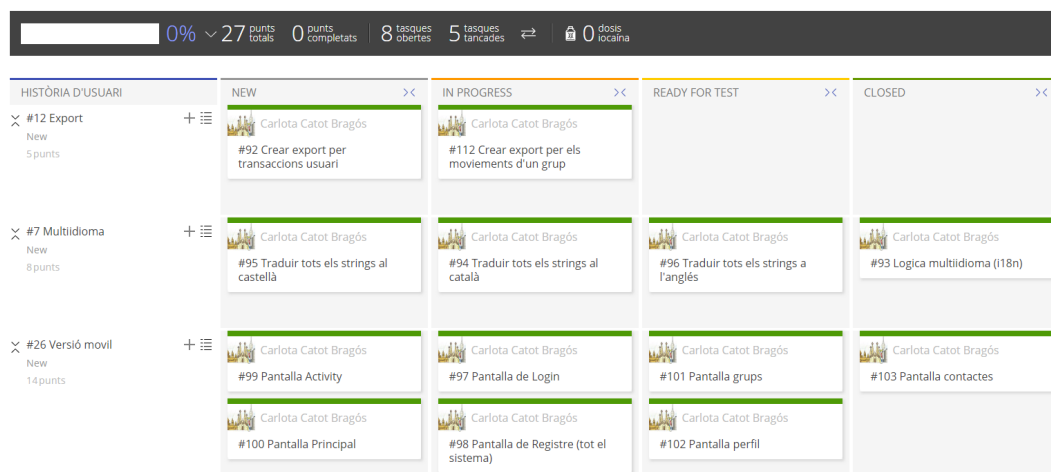


Figura 14.15: Panell de tasques de l'sprint 6

14.7 Sprint 7

Donat el backlog després del sisè sprint (imatge 14.16), pel setè sprint s'assignaran totes les històries d'usuari restants, **Millores - 15 punts** i **Animacions i disseny - 6 punts** i una durada de 4 setmanes (del 20 de maig del 2019 al 17 de juny del 2019), les tasques assignades a aquest sprint (figura 14.17) no s'ha assignat massa tasques, ja que totes les tasques consten de fer les millores a partir del testing o fer l'aplicació més atractiva pels usuaris. Tota la informació de l'Sprint es pot consultar al taiga [43]

Tasques

1. **Pantalla Help:** Creació de la pantalla d'ajuda. Per tal de realitzar aquesta tasca s'ha realitzat una pantalla d'ajuda on s'expliquen les funcionalitats de cada pantalla.
2. **Canvis testing:** Realització dels canvis que s'han demanat al testing. Per tal de realitzar aquesta tasca es deixarà provar l'aplicació a diversos usuaris i es corregiran els errors que aquests trobin.
3. **Canvis disseny testing:** Realització dels canvis que s'han demanat al testing. Per tal de realitzar aquesta tasca es deixarà provar l'aplicació a diversos usuaris i es corregiran els errors que aquests trobin.

4. **Loading pàgines:** Implementació d'animacions de loading a les pàgines. Per tal de realitzar aquesta tasca caldrà modificar tot el sistema per afegir un spinner quan les dades s'estan carregant.

Resum de les hores invertides al desenvolupament Front-End durant l'Sprint

Tasca	Hores previstes	Hores reals
Pantalla help	3	3
Loading pàgines	3	3
Hores fora de tasca		6
TOTAL		12

Taula 14.12: Hores invertides al desenvolupament Front-End durant l'Sprint 7

Resum de les hores invertides en millores durant l'Sprint

Tasca	Hores previstes	Hores reals
Canvis testing	15	15
Canvis disseny testing	15	10
TOTAL		25

Taula 14.13: Hores invertides en millores durant l'Sprint 7

TFG - COINK BACKLOG



Figura 14.16: Backlog després de l'sprint 6

TFG - COINK SPRINT 7 20 MAIG 2019-17 JUNY 2019

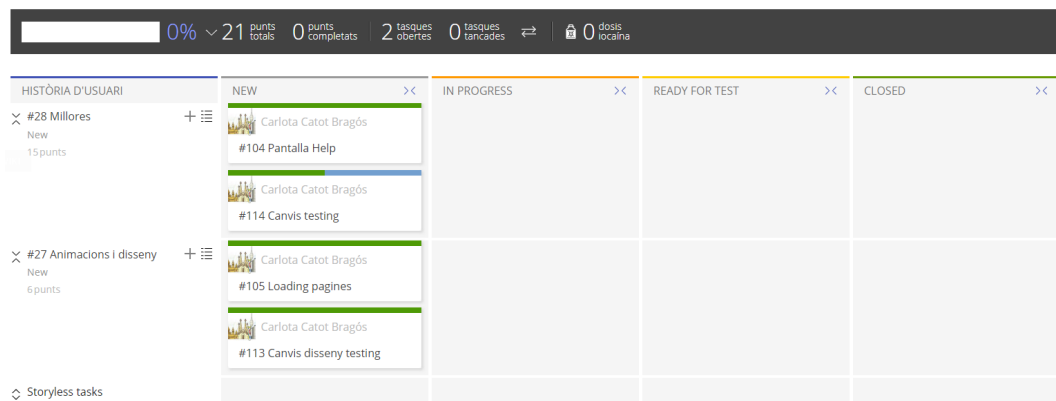


Figura 14.17: Panell de tasques de l'sprint 7

14.7.1 Resum d'hores invertides

Desenvolupament Back-End

Sprint	Hores previstes	Hores reals
Sprint 1	18	15
Sprint 2	10	13
Sprint 3	36	39
Sprint 4	29	33
Sprint 5	57	75
TOTAL	150	175

Taula 14.14: Resum d'hores invertides en el desenvolupament de Back-End

Desenvolupament Front-End

Sprint	Hores previstes	Hores reals
Sprint 1	4	9
Sprint 2	9	13
Sprint 3	42	57
Sprint 4	14	21
Sprint 5	62	78
Sprint 6	63	59
Sprint 7	6	12
TOTAL	200	249

Taula 14.15: Resum d'hores invertides en el desenvolupament de Front-End

Millores

Sprint	Hores previstes	Hores reals
Sprint 7	30	25
TOTAL	30	25

Taula 14.16: Resum d'hores invertides en millores

14.8 Configuracions finals AWS

Un cop acabat tot el desenvolupament es presenta la configuració final d'AWS mitjançant unes captures de pantalla de la consola d'AWS.

14.8.1 API Gateway

A la imatge 14.18 podem veure la consola de l'API Gateway on es troba el sistema de Coink. Veiem com a exemple la funció de GET /balance, tot i que tots els endpoints tenen la mateixa estructura, al botó de *test* es on es realitzen les proves que s'expliquen al capítol d'estratègia de testing i validació. Destacar que hi ha informació ocultada perquè són claus de configuració.

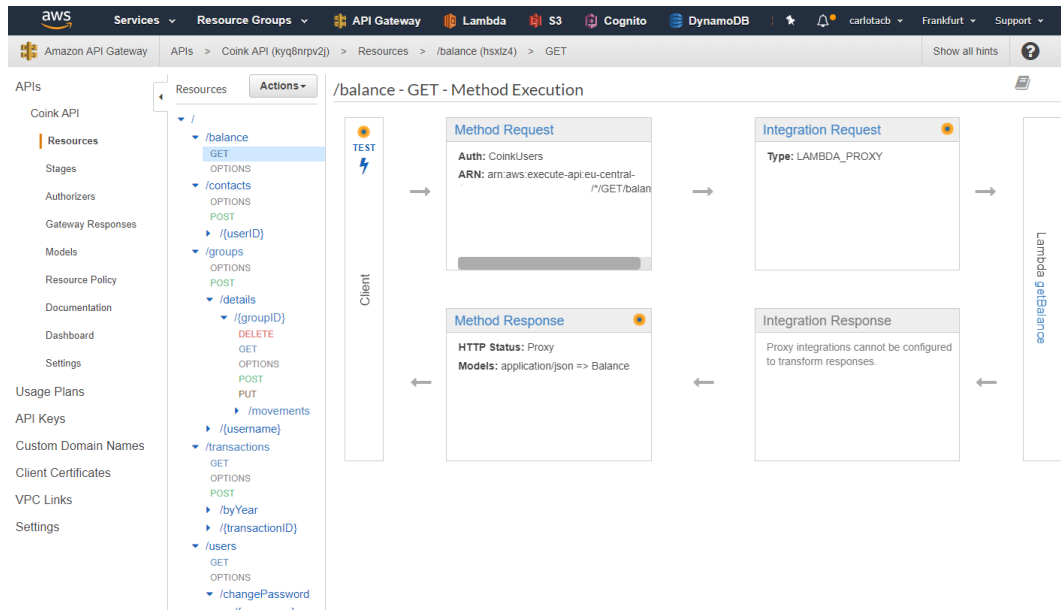


Figura 14.18: Configuració final API Gateway

14.8.2 Lambdas

Pel que fa a la configuració de les funcions lambda es presenten 3 imatges.

A la primera imatge 14.19 podem veure que s'han programat 29 funcions que corresponen als 29 endpoints de l'aplicació.

Function name	Description	Runtime	Code size	Last modified
resendCodeLambda		Node.js 10.x	536.7 kB	2 days ago
putUpdateInformation		Node.js 10.x	6.0 MB	2 days ago
loginLambda		Node.js 10.x	536.7 kB	2 days ago
registerLambda		Node.js 10.x	536.9 kB	2 days ago
getUpdateUserInformation		Node.js 10.x	6.0 MB	2 days ago
logoutLambda		Node.js 10.x	536.5 kB	2 days ago
confirmRegistrationLambda		Node.js 10.x	536.7 kB	2 days ago
changePasswordLambda		Node.js 10.x	6.0 MB	2 days ago
getUsers		Go 1.x	6.6 MB	2 days ago
getGroupInformation		Go 1.x	6.6 MB	6 hours ago

Figura 14.19: Resum de les funcions que hi ha a les lambdes

A la segona imatge 14.20 podem veure la configuració d'una funció programada en Go-Lang, com es pot veure, i s'ha explicat anteriorment, el codi es penja a través d'un fitxer zip que conté l'executable de la funció i com a handler s'especifica el nom d'aquest executable. A la part superior es veu un diagrama que indica a quins recursos d'AWS és connecta aquesta lambda (els recursos es posen automàticament en realitzar les diferents configuracions).

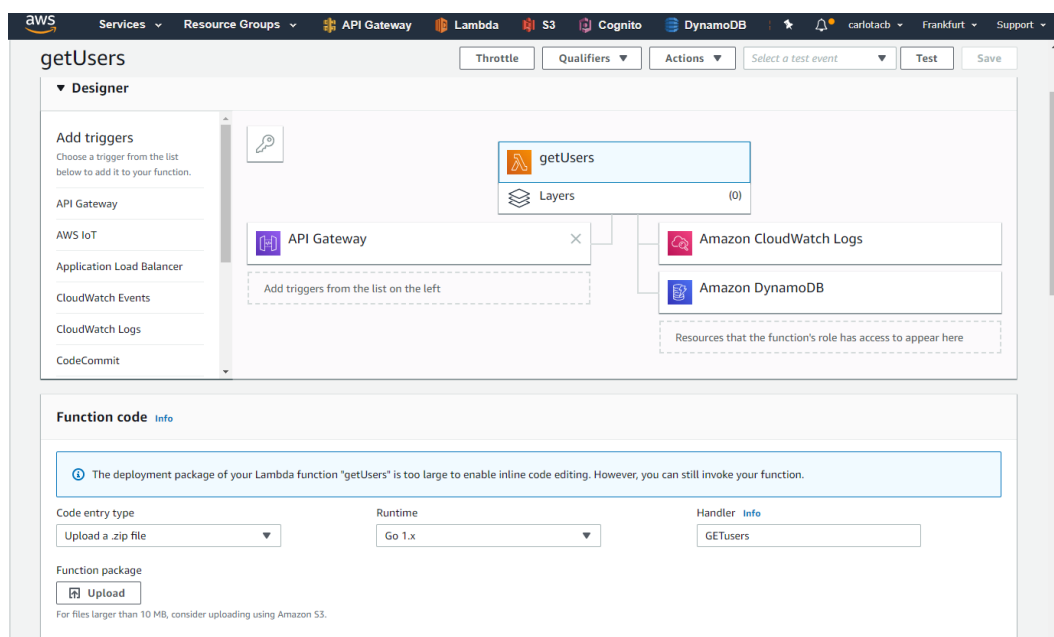


Figura 14.20: Exemple de configuració d'una funció lambda programada en Go

A la tercera i última imatge 14.21 es presenta una funció programada en JavaScript, en aquest cas podem veure, igual que per la imatge anterior un diagrama dels recursos utilitzats (en aquest cas són més perquè es connecta al sistema de cognito). Però com a diferència podem veure que el codi d'aquesta funció es pot editar directament des de la consola el que fa que el procés d'edició de text sigui més ràpid.

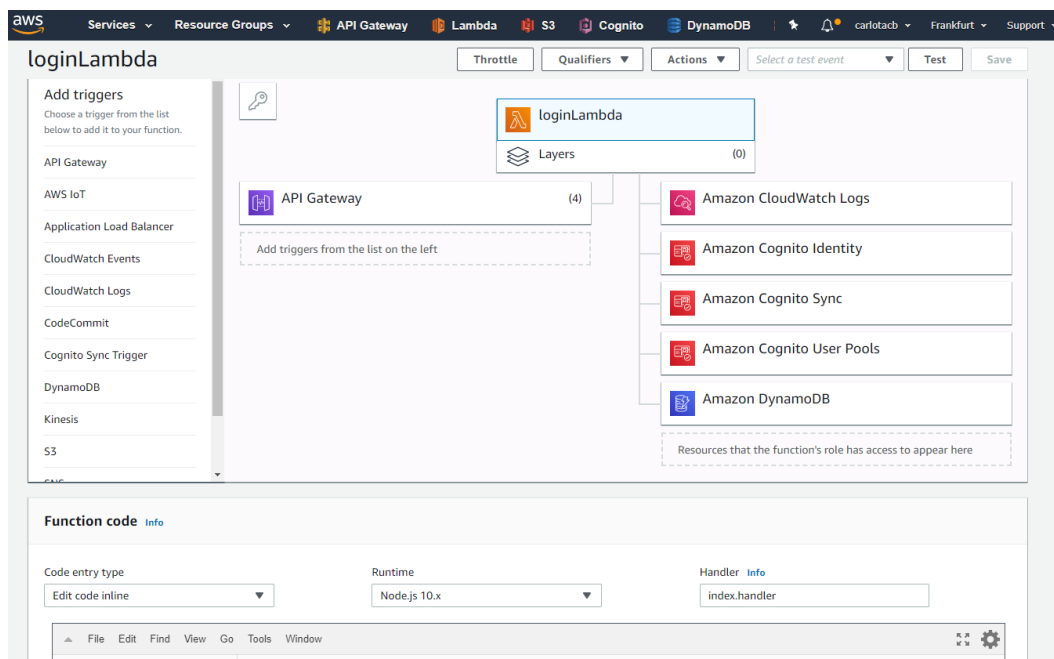
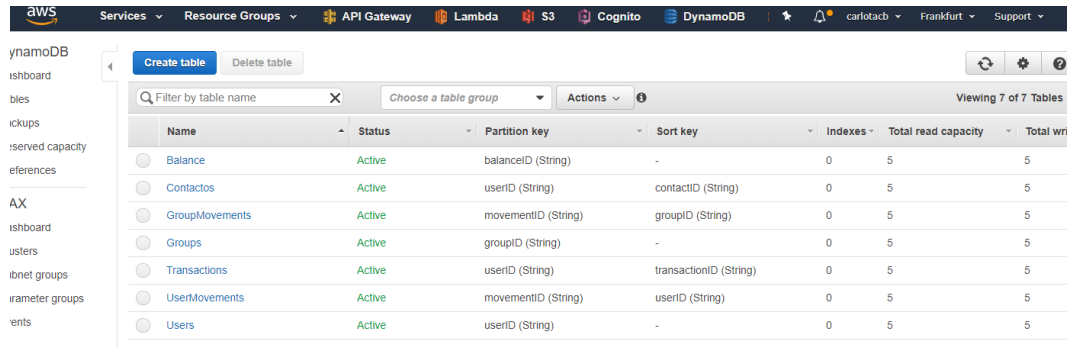


Figura 14.21: Exemple de configuració d'una funció lambda programada en JavaScript

14.8.3 DynamoDB

Per DynamoDB es presenta la imatge 14.22 que és bastant autoexplicativa, ja que es poden veure les 7 taules que s'utilitzen al sistema amb les claus primàries de cadascuna.

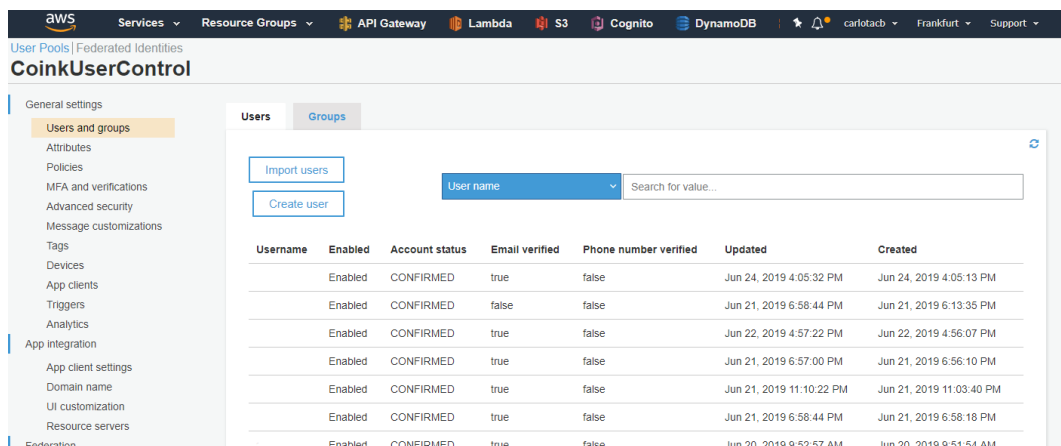


Name	Status	Partition key	Sort key	Indexes	Total read capacity	Total write capacity
Balance	Active	balanceID (String)	-	0	5	5
Contactos	Active	userID (String)	contactID (String)	0	5	5
GroupMovements	Active	movementID (String)	groupID (String)	0	5	5
Groups	Active	groupID (String)	-	0	5	5
Transactions	Active	userID (String)	transactionID (String)	0	5	5
UserMovements	Active	movementID (String)	userID (String)	0	5	5
Users	Active	userID (String)	-	0	5	5

Figura 14.22: Configuració final DynamoDB

14.8.4 Cognito

Per ultim es presenta la consola de Cognito a la imatge 14.23, s'han tapat els Usernames per tal de donar privacitat als usuaris que estan registrats a l'aplicació. En aquesta consola es poden habilitar i deshabilitar usuaris, els camps d'un usuari es poden variar sempre que es vulgui a excepció dels camps obligatoris que s'han hagut de marcar desde un inici.



Username	Enabled	Account status	Email verified	Phone number verified	Updated	Created
	Enabled	CONFIRMED	true	false	Jun 24, 2019 4:05:32 PM	Jun 24, 2019 4:05:13 PM
	Enabled	CONFIRMED	false	false	Jun 21, 2019 6:58:44 PM	Jun 21, 2019 6:13:35 PM
	Enabled	CONFIRMED	true	false	Jun 22, 2019 4:57:22 PM	Jun 22, 2019 4:56:07 PM
	Enabled	CONFIRMED	true	false	Jun 21, 2019 6:57:00 PM	Jun 21, 2019 6:56:10 PM
	Enabled	CONFIRMED	true	false	Jun 21, 2019 11:10:22 PM	Jun 21, 2019 11:03:40 PM
	Enabled	CONFIRMED	true	false	Jun 21, 2019 6:58:44 PM	Jun 21, 2019 6:58:18 PM
	Enabled	CONFIRMED	true	false	Jun 20, 2019 9:59:57 AM	Jun 20, 2019 9:51:54 AM

Figura 14.23: Configuració final Cognito

14.9 Release burndown and velocity chart

En aquesta secció podem veure dues gràfiques que mostren el progrés del projecte a través de les històries d'usuari.

En primer lloc podem veure a la *Release Burndown Chart* (figura 14.24) que el projecte s'ha realitzat de manera pràcticament lineal pel que fa a la realització de les històries d'usuari, explicant més en detall, aquesta gràfica mostra el progrés de les històries realitzades indicant per a cada Sprint (Eix X) el nombre de punts d'històries d'usuari (definitos a l'inici del projecte i redefinitos a l'Sprint 2) el nombre d'històries d'usuari restants un cop acabat cada Sprint, és per això que es comença a comptar en 0 per tal d'indicar el nombre total de punts d'història d'usuari a realitzar durant el projecte.

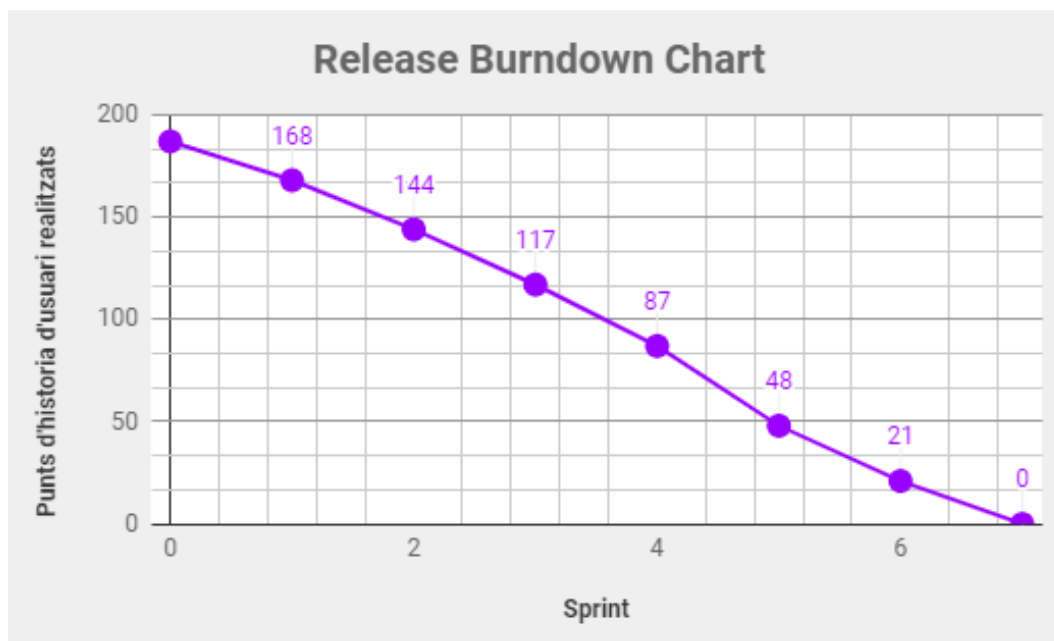


Figura 14.24: Gràfica de l'evolució dels punts d'histories d'usuari

Pel que fa a la *Velocity Chart* (figura 14.25) podem veure en forma de grafica de barres els punts d'història d'usuari realitzats per cada sprint. Aquesta gràfica ens mostra que a mesura que ha anat avançant el desenvolupament s'ha anat realitzant més punts per Sprint de manera lineal de l'1 al 4, a l'Sprint 5 s'ha agafat molta més càrrega, però els dos últims han sigut més relaxats.

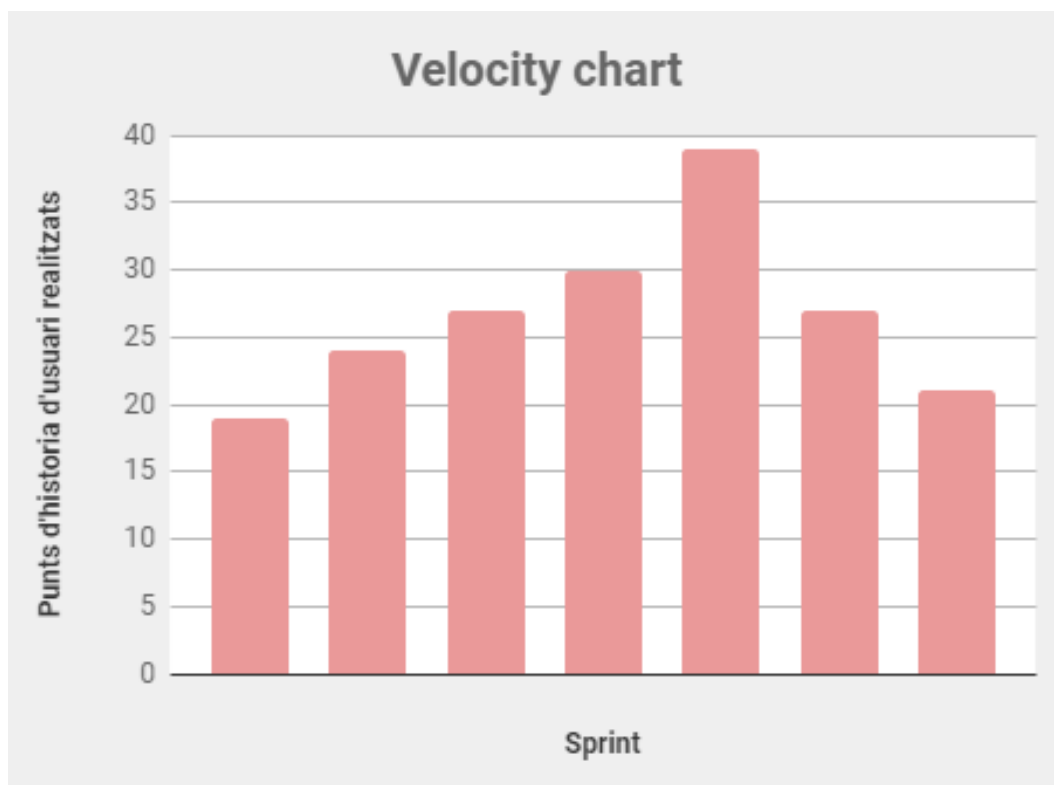


Figura 14.25: Gràfica de la velocitat de realització dels punts d'histories d'usuari

Capítol 15

Estrategia de treball i desenvolupament

Com a estratègia de treball durant el projecte s'ha utilitzat el mètode Agile per la realització de les diferents tasques, i que s'ha explicat al capítol diari d'execució. Però per tal de completar aquestes s'han seguit unes estratègies de treball i de desenvolupament explicades a continuació.

Estratègia de treball

Com a estratègia de treball per dissenyar i desenvolupar l'aplicació per cadascuna de les diferents funcionalitats es va pensar en primer lloc la interfície gràfica per tal de posteriorment definir els endpoints i d'aquesta manera desenvolupar ambdues parts. Per tal de definir la interfície gràfica es va utilitzar el mètode de dibuixar amb paper i boli les diferents pantalles i posteriorment es van desenvolupar, pel que fa als endpoints, per tal d'anar-los definint s'ha utilitzat Swagger, la versió definitiva del sistema es pot consultar a l'URL: <https://app.swaggerhub.com/apis-docs/carlotacb/Coink-API/2.0#/>.

Estratègia de desenvolupament

Per tal d'explicar l'estratègia de desenvolupament, el dividirem en Front-End i Back-End.

Pel que fa al Front-End per tal de desenvolupar tota la web s'ha utilitzat React.js, un framework de JavaScript, juntament amb els components de disseny d'AntDesign. Mitjançant Antdesign s'han pogut completar moltes de les funcionalitats d'una manera més senzilla tot i que s'ha hagut d'aprendre i fer una cerca per cadascun dels elements que es necessitaven. El software utilitzat pel desenvolupament ha estat WebStorm, un editor de text de la companyia JetBrains que està enfocat al desenvolupament web.

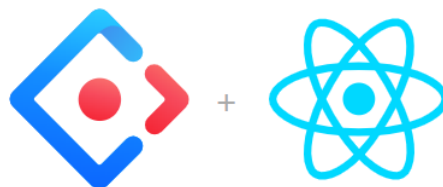


Figura 15.1: Eines utilitzades pel desenvolupament del Front-End, AntDesign + React.js

Pel que fa al Back-End s'han utilitzat diversos llenguatges de programació, ja que el desenvolupament d'aquest ha patit diferents etapes. Per cadascun dels endpoints definits al swagger (programats en yalm) i presentats anteriorment, s'ha creat una funció lambda, totes les funcions que tenen relació amb la connectivitat a Cognito han estat programades en NodeJS, mentre que la resta de funcions han estat programades en Golang. Això, com s'explica a l'Sprint Backlog ve donat perquè en un primer moment totes les funcions de

cognito estaven programades al Front-End, però com per tal de fer les peticions és necessari tenir en el codi les credencials de cognito es va migrar el codi a diferents lambdes, tot i que no va funcionar a la primera es va decidir no traduir-ho a Go per tal de no perdre temps. El software utilitzat pel desenvolupament ha estat GoLand, un editor de text de la companyia JetBrains que està enfocat al desenvolupament amb Golang.



Figura 15.2: Llenguatges utilitzats pel desenvolupament del Back-End, Golang i NodeJS

Funcionalitats a destacar

Totes les funcionalitats han sigut explicades prèviament al capítol de disseny del sistema o al de diari d'execució, però es vol destacar un parell de detalls d'implementació de diverses funcionalitats.

- Totes les pantalles disposen de la versió mòbil per tal de complir les propietats d'una WebApp.
- Per totes les taules que es presenten a les diferents pantalles s'ha implementat un sistema de paginació, de sorting, de cerca i de filtre.
- Per tot el sistema s'ha implementat un servei de multiidioma que permet utilitzar l'aplicació en català, en castellà o en anglès.
- S'ha creat una pantalla d'ajuda, per tal de solucionar els dubtes de l'usuari en el cas que la interfície no sigui prou bona.
- S'han utilitzat elements visuals per tal d'indicar les despeses (en vermell), les despeses (en verd) i s'han posat icones pels botons.
- Totes les pantalles tenen la funcionalitat de loading per tal que l'usuari sigui conscient que les dades s'estan carregant.
- Quan la sessió de l'usuari es caduca, es tanca la sessió de manera automàtica.

Capítol 16

Estrategia de testing i validació

Pel testing i la validació de les funcionalitats de l'aplicació s'han utilitzat diferents sistemes, pel cas del Back-End s'han realitzat testos directament fent les proves a la consola d'AWS i posteriorment també amb el sistema de Postman, i pel cas del Front-End s'ha demanat a companys de carrera i familiars que provin l'aplicació per tal d'extreure errors. (que també testeja el Back-End indirectament)

16.1 Testing AWS

Pel testing dels endpoints s'ha utilitzat la consola de l'API Gateway on es poden introduir les dades necessàries per fer la petició a l'endpoint, a la imatge 16.1 es pot veure el cas del test que s'ha realitzat per l'endpoint de GET /transactions introduint en el header el token d'un usuari actiu per tal de completar l'autenticació de l'API Gateway i el nom d'usuari de l'usuari actiu en forma de query parameter. Per comprovar els codis d'error també s'han realitzat proves amb valors incorrectes per tal de veure els diferents comportaments de l'API.

← Method Execution /transactions - GET - Method Test

Make a test call to your method with the provided input

Path

No path parameters exist for this resource. You can define path parameters by using the syntax `{myPathParam}` in a resource path.

Query Strings

`{transactions}`

username=carlota

Headers

`{transactions}`

Authorization:eyJraWQlOUI5K1pxaGQzMDI1VXUwUko5SjJlTmxmSks5S3FMemtUOXdcL1RQeVZOZU0yaz0iLjJhbGciOiJSUzI1NiJ9.eyJzdWIiOiIwZTVjYzMyZC1mNDA5LTRhYzQyYjg1

Stage Variables

No [stage variables](#) exist for this method.

Client Certificate

No client certificates have been generated.

Request Body

Request Body is not supported for GET methods.

Request: /transactions?username=carlota

Status: 200

Latency: 31 ms

Response Body

```
[
  {
    "userID": "carlota",
    "transactionID": "cffc5057-b89d-4413-b7f5-21d91391d6f0",
    "title": "Sopar",
    "description": "Sopar al Vapiano",
    "category": "Alimentacio i Llar",
    "date": "04/06/2019",
    "amount": -18,
    "expense": true
  },
  {
    "userID": "carlota",
    "transactionID": "ec4b51fd-eba4-4a3c-8352-e26478ccdc6b",
    "title": "Gasolina",
    "description": "Omplir el diposit del coche",
    "category": "Vehicle",
    "date": "03/06/2019",
    "amount": -30,
    "expense": true
  },
  {
    "userID": "carlota",
    "transactionID": "ef874c23-c941-475c-b9e7-c34f6e768615",
    "title": "Regal cumple",
    "description": "Diners avis",
    "category": "Efectiu",
    "date": "20/06/2019",
    "amount": 50,
    "expense": false
  }
]
```

Figura 16.1: Exemple de testing per l'endpoint de GET /transactions

16.2 Validació amb usuaris reals

En aquest apartat es detalla el feedback que s'ha rebut per part dels usuaris que han provat l'aplicació, i per tal de no canviar les seves paraules s'ha escrit en l'idioma que s'ha fet el feedback. Tot el que s'ha rebut com a feedback s'ha modificat per al correcte funcionament de l'aplicació, algunes de les versions definitives de les imatges que és mostraran en aquest apartat es poden veure al capítol de funcionalitats de l'aplicació.

Andreu Gallofré - Estudiant de la FIB

1. Les gràfiques d'expenses estàn malament, només surten la meitat de les categories (figura 16.2).
2. La data no s'autocompleta quan vas a ficar una expense/income amb la data del dia d'avui, has de fer click.
3. Les transaccions no tenen cap tipus de filtre per data.
4. A l'apartat de contactes s'hauria de treure el mail dels usuaris, ja que això és un leak de seguretat (aquest error es pot veure a la figura 16.3, per un tema de confidencialitat s'ha tapat una part de l'e-mail a la imatge).

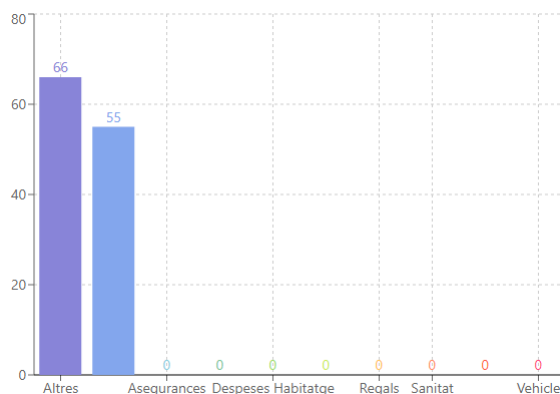


Figura 16.2: Gràfica de gastos en la versió dolenta

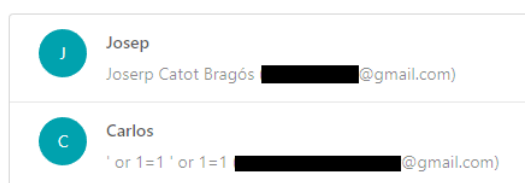


Figura 16.3: Fragment on es veu l'error dels e-mails

Carlos Roldán, Didac Plaza i Juan Miguel de Haro - Graduats en Enginyeria Informàtica per la FIB

1. Hay infinitos logs en la consola (axios.js,register.js,etc) que dan bastante info.
2. Puedes registrarte con un balance no numérico como por ejemplo "asdf". Te lo cuenta como 0 pero igual no debería dejarte (figura 16.4).
3. Hay un problema **importante** con el tema de añadir ingresos y gastos. Puedes poner números negativos, lo que hace que puedas tener gastos que te dan dinero e ingresos que te lo quitan, haciendo que no haya diferencia entre ellos. No solo eso sino que afecta al resto de info y gráficas, por ejemplo he llegado a tener una flechita hacia arriba en

verde de -20 euros. Además aunque en las tablas salían gastos de por ejemplo -5 y +5, en las gráficas salían ambos positivos (supongo que porque están limitadas, mientras que el input no). De manera que tenías un gasto de 0 (porque se cancelan), que en la gráfica se mostraba como 10. Todo esto se soluciona si bloqueas los negativos en el input de Gastos e Ingresos.

4. Puedes añadir a todos los usuarios, que entiendo que es una funcionalidad a lo red social. El problema es que displayea los mails de todos los usuarios que es una fiesta para los spambots (figura 16.3).

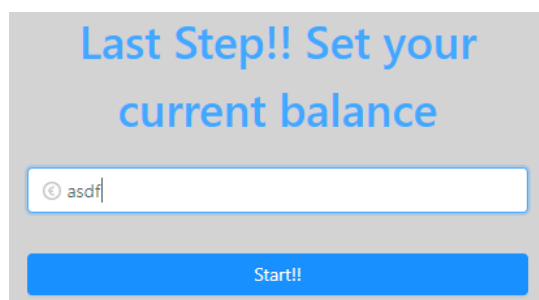


Figura 16.4: Fragment on es veu que es pot afegir un balanç que no sigui numèric

Josep Catot - Estudiant de Ciències Polítiques a la UPF

1. Quan esborres un usuari de contactes i només tenies un usuari, es queda fent loading fins que recareges la pàgina.
2. Posar un color blau més fort a la frase de la pantalla home, ja que aquest color és massa clar pel fons clar que té, és un to que es veu bé per fons més foscos (figura 16.5).
3. Quan vas a crear un grup no hi ha autocomplete amb els contactes que te l'usuari.
4. Quan et queda només una transacció i la borres, no desapareix fins que recarregues la pàgina.
5. Quat et queda un grup i l'esborres no desapareix fins que recarregues la pàgina.

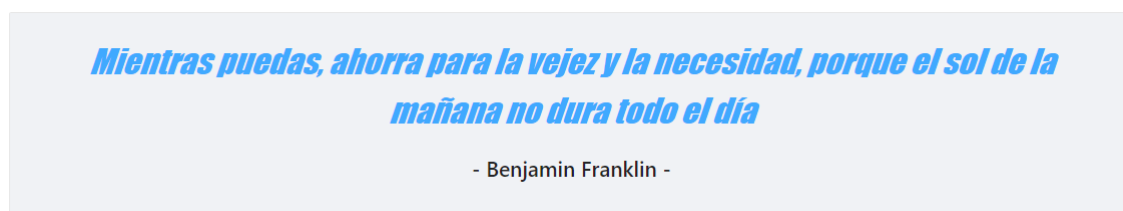


Figura 16.5: Fragment on es veu el color clar de la frase

Cristina Bragós - Mastèr IBM en software sistemes HOST

1. Afegir la categoria Efectiu per les Income Transactions (a la figura 16.6 es poden veure les categories que hi havia quan es va fer el testing).
2. Es poden posar transaccions del futur, s'hauria de bloquejar les dates futures (figura 16.7).

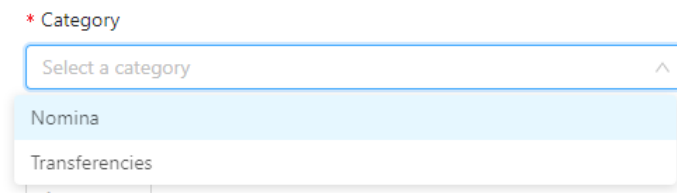


Figura 16.6: Fragment on es veuen les categories per les transaccions Income

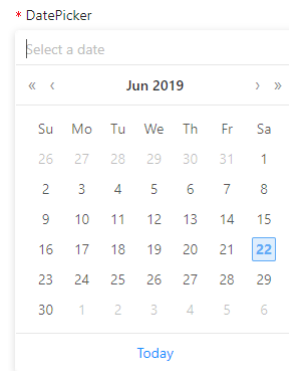


Figura 16.7: Fragment on es veuen que es poden afegir dates futures

Xavier Bragós - Consultora CTTI, departament de justícia de la Generalitat de Catalunya

No s'ha trobat pràcticament cap error, el sistema és molt intuïtiu, només afegiria un punt a modificar: Afegir popups en passar el ratolí damunt dels botons d'acció que són només icones.

Capítol 17

Obstacles

Després de completar el desenvolupament del projecte i partint dels possibles obstacles que es podien trobar, els obstacles a destacar durant el desenvolupament i les solucions d'aquests es presenten a continuació.

17.1 Existència de coink.com

El nom de l'aplicació es va decidir durant l'assignatura de GEP, però en aquell moment (error important) no es va fer una cerca per tal de comprovar l'existència d'alguna cosa amb aquest nom. Més endavant, al moment d'obtenir un domini es va cercar *Coink* i es va trobar la web de la imatge 17.1 en ser una pàgina web que encara està amb construcció i veient que ja s'havia fet el logo de l'aplicació i no es trobava un altre nom (per falta de creativitat), amb l'aprovació de la directora es va decidir mantenir el nom de Coink per aquest projecte, ja que en ser només un TFG no repercuteix en termes legals. Durant la setmana d'entrega de la memòria s'ha accedit a la web i s'ha comprovat que encara està en manteniment.



Figura 17.1: Pàgina web del dia que es va descobrir [44]

17.2 Temps

El temps és un obstacle que ja es preveia des de l'inici del projecte, tot i que totes les funcionalitats han estat acabades, no de manera òptima, per tal de completar un MVP complint els objectius i dins del període establert s'ha decidit eliminar funcionalitats que potser d'un inici es volien incorporar com és el cas de la *Multidivisa*, per tant passarà a ser un punt de possibles funcionalitats futures, ja que no era una funcionalitat bàsica i d'aquesta manera s'ha completat la resta de funcionalitats necessàries a temps.

17.3 Funcionament Amazon Web Services

Tot el sistema d’AWS era una cosa nova que s’ha hagut d’aprendre de 0, això ha sigut un obstacle pel que fa a la motivació, ja que en veure que no sortien les coses ha costat més avançar ràpidament i en moltes ocasions ha causat un coll d’ampolla. Finalment i després del desenvolupament de tot el projecte es pot considerar que la desenvolupadora del projecte és bastant experta en els sistemes AWS utilitzats pel projecte.

Capítol 18

Conclusions

Després d'haver explicat tots els detalls importants sobre el disseny i desenvolupament de *Coink*, per a les conclusions s'analitzaran si s'han assolit els objectius marcats a l'inici del projecte, si s'han assolit les competències tècniques definides a l'inici. Per acabar, es planteja un possible futur pel projecte i per últim es farà una reflexió sobre l'aprenentatge personal de l'estudiant durant el desenvolupament del projecte.

18.1 Assoliment d'objectius del projecte

Pels objectius del projecte han estat definits al capítol d'objectius, s'han assolit de manera correcta tots ells.

L'objectiu principal del projecte estava resumit així: *dissenyar i desenvolupar una aplicació prou bona per poder ajudar als usuaris a contralorar les despeses*. Després de tot l'explicat durant aquest document es pot dir que s'ha assolit correctament.

Pel que fa als subobjectius també es pot dir que s'han assolit correctament. A continuació es justifica l'assoliment de cadascun d'aquests:

- **Complir les propietats de WebApp:** com s'ha destacat dins el capítol d'estrategia de treball i desenvolupament, l'aplicació consta d'una versió web i una versió mòbil per tant compleix les propietats d'una WebApp.
- **Gestió de tràmits complerta:** al capítol de disseny del sistema, on s'expliquen les funcionalitats de l'aplicació podem veure que aquest disposa d'una gestió de tràmits complerta.
- **Gestió d'usuaris:** al capítol de disseny del sistema, on s'expliquen les funcionalitats de l'aplicació podem veure que aquest disposa d'una gestió d'usuaris.
- **Gestió de grups:** al capítol de disseny del sistema, on s'expliquen les funcionalitats de l'aplicació podem veure que aquest disposa d'una gestió de grups.
- **Bona arquitectura:** al capítol de disseny del sistema, s'expliquen amb detall l'arquitectura que s'ha utilitzat per les diferents parts del sistema, així com tot el sistema d'AWS utilitzat.

Per últim l'objectiu personal, també s'ha assolit amb èxit. Aquest es troba explicat a l'última secció d'aquest capítol amb les conclusions personals.

18.2 Assoliment competències tècniques del projecte

Les competències tècniques definides pel projecte han estat les següents:

- CES1.1: Desenvolupar, mantenir i avaluar sistemes i serveis software complexos i/o crítics. **UNA MICA**

- CES1.2: Donar solució a problemes d'integració en funció de les estratègies, dels estàndards i de les tecnologies disponibles. **BASTANT**
- CES1.3: Identificar, avaluar i gestionar els riscos potencials associats a la construcció de software que es poguessin presentar. **UNA MICA**
- CES1.5: Especificar, dissenyar, implementar i avaluar bases de dades. **UNA MICA**
- CES1.7: Controlar la qualitat i dissenyar proves en la producció de software. **EN PROFUNDITAT**
- CES2.1: Definir i gestionar els requisits d'un sistema software. **BASTANT**

Pel que fa a les competències que s'han determinat com *UNA MICA*, podem dir que totes s'han assolit.

- Per la competència 1.1 podem dir que s'han desenvolupat diverses funcionalitats que s'han anat mantenint durant tot el desenvolupament, ja que s'han hagut d'anar millorant a partir dels diferents obstacles i proves realitzades.
- Per la competència 1.3 de la mateixa manera que per la competència anterior, s'han anat gestionant els diferents problemes que s'han trobat tant en l'àmbit de codi com en l'àmbit de disseny durant el procés de construcció de l'aplicació.
- Per últim, la competència 1.5 s'ha assolit perfectament, ja que s'han especificat, dissenyat i implementat la bases de dades mitjançant DynamoDB.

Pel que fa a les competències que s'han determinat com *BASTANT*, podem dir que totes s'han assolit.

- Per la competència 1.2 s'ha realitzat la implementació del sistema utilitzant les tecnologies més adients per cada funcionalitat (definit al capítol d'estratègia de treball i desenvolupament).
- Per la competència 2.1, com s'ha vist anteriorment i també s'ha dedicat un capítol sencer s'han definit els requisits del sistema d'una manera molt acurada, això es pot veure al capítol de requisits.

Per últim, la competència 1.7 que s'ha determinat com a *EN PROFUNDITAT*, podem dir que s'ha assolit de manera correcta, ja que en tot moment s'han realitzat diverses proves per tal d'assegurar tenir un software de qualitat seguint els requisits i els objectius marcats.

18.3 Treball futur

Aquesta aplicació podria tenir molt futur, principalment es podria completar amb funcionalitats que no s'han implementat per falta de temps com és la Multidivisa o les notificaciones. En l'hipotètic cas que aquesta aplicació tingui un futur, els primers next steps que seguiria són els següents:

1. Afegir connectivitat amb bancs per tal de posar de manera automàtica les despeses realitzades a través del banc.
2. Gestionar els contactes per invitació.
3. Realitzar divisió de despeses desigual en un grup
4. Afegir l'opció de multidivisa amb el canvi que té actualment.
5. Crear aplicacions mòbil per tal d'oferir totes les funcionalitats en els diferents dispositius.

Posteriorment es completaria amb noves funcionalitats de manera que l'aplicació es pugui adaptar a les necessitats dels usuaris escoltant les seves peticions i queixes.

18.4 Conclusions personals

Aquest ha estat un treball molt gratificant en l'àmbit personal, ja que s'han aconseguit uns objectius, en la meva opinió, bastant ambiciosos tenint en compte el coneixement inicial de les tecnologies a l'inici d'aquest. Cal destacar que abans d'aquest projecte personalment em considerava una desenvolupadora bàsicament enfocada al Front-End i a dia d'avui, després de l'experiència de dissenyar i desenvolupar tot un sistema em puc considerar una mica més desenvolupadora Full-Stack.

Durant el projecte s'ha realitzat una API i s'ha après com utilitzar els recursos d'AWS dels que es tenia coneixement nul, també he après a com realitzar una bona interfície d'usuari. Tots els coneixements adquirits durant aquest projecte fan que el patiment durant aquest en algunes ocasions amb els diferents obstacles que s'han anat trobant, valgui la pena.

Bibliografia

- [1] Blog sobre l'estalvi a les llars a Europa pel diari l'economista [cons. el dia 22/03/2019]
<http://www.expansion.com/ahorro/2018/10/22/5bcd0f1e5fdeaa2678b469c.html>
- [2] Imatge extreta del blog de l'economista del punt anterior, amb la Font original del CNMV butlletí trimestral III/2018
- [3] Per tal de fer l'estudi de mercat d'aquesta aplicació (Splitwise) es va descarregar a través de l'App Store [cons. el dia 13/01/2019]
<https://www.apple.com/es/search/splitwise?src=globalnav>
- [4] També es va cercar i utilitzar la seva pàgina web [cons. el dia 20/02/2019]
<https://www.splitwise.com/>
- [5] Igual que splitwise, Tricount també es va cercar i descarregar a l'App Store [cons. el dia 20/02/2019]
<https://www.apple.com/es/search/tricount?src=serp>
- [6] També es va cercar i utilitzar la seva pàgina web [cons. el dia 22/02/2019]
<https://www.tricount.com/es/>
- [7] Igual que per totes les aplicacions nomenades abans, Expense manager - Money tracker es va cercar i descarregar a l'App Store [cons. el dia 22/02/2019]
<https://itunes.apple.com/us/app/expense-manager-money-tracker/id1104167285?mt=8>
- [8] Igual que per totes les aplicacions nomenades abans, iSpending es va cercar i descarregar a l'App Store [cons. el dia 22/02/2019]
<https://itunes.apple.com/us/app/ispending-expense-tracker/id484100875?mt=8>
- [9] Igual que per totes les aplicacions nomenades abans, Fintonic es va cercar i descarregar a l'App Store [cons. el dia 22/02/2019]
<https://itunes.apple.com/es/app/fintonic/id672220319?mt=8>
- [10] Per Fintonic també es va cercar i utilitzar la seva pàgina web [cons. el dia 22/02/2019]
<https://www.fintonic.com/es-ES/inicio/>
- [11] Eina per realitzar el diagrama de Gantt [cons. en múltiples ocasions]
<https://www.teamgantt.com/>
- [12] Blog on s'explica com definir els criteris d'acceptació d'una aplicació [cons. el dia 08/03/2019]
<https://samuelcasanova.com/2017/11/criterios-de-aceptacion/>
- [13] Eina per realitzar els diagrames de disseny del sistema [cons. en múltiples ocasions]
<https://www.draw.io/>
- [14] Blog on s'expliquen els patrons de disseny que utilitzen els microserveis d'AWS [cons. el dia 11/06/2019]
<https://www.jeremydaly.com/serverless-microservice-patterns-for-aws/>
- [15] JavaScript Design Patterns - Addy Osmani. O'Reilly, 2012

- [16] Fragment del llibre *JavaScript Design Patterns* - Addy Osmani. O'Reilly, 2012 on s'explica el patró constructor [cons. el dia 20/06/2019]
<https://addyosmani.com/resources/essentialjsdesignpatterns/book/#constructorpatternjavascript>
- [17] Fragment del llibre *JavaScript Design Patterns* - Addy Osmani. O'Reilly, 2012 on s'explica el patró observador [cons. el dia 20/06/2019]
<https://addyosmani.com/resources/essentialjsdesignpatterns/book/#observerpatternjavascript>
- [18] Fragment del llibre *JavaScript Design Patterns* - Addy Osmani. O'Reilly, 2012 on s'explica el patró prototip [cons. el dia 20/06/2019]
<https://addyosmani.com/resources/essentialjsdesignpatterns/book/#prototypepatternjavascript>
- [19] Fragment del llibre *JavaScript Design Patterns* - Addy Osmani. O'Reilly, 2012 on s'explica el patró fatxada [cons. el dia 20/06/2019]
<https://addyosmani.com/resources/essentialjsdesignpatterns/book/#facadepatternjavascript>
- [20] Fragment del llibre *JavaScript Design Patterns* - Addy Osmani. O'Reilly, 2012 on s'explica el patró factoria [cons. el dia 20/06/2019]
<https://addyosmani.com/resources/essentialjsdesignpatterns/book/#factorypatternjavascript>
- [21] Panell de l'Sprint 1 del Taiga [cons. en múltiples ocasions]
<https://tree.taiga.io/project/carlotacatot-tfg-coink/taskboard/sprint-1-14149>
- [22] Tutorial d'AWS per la realització de l'Sprint 1 [cons. el 05/03/2019]
<https://aws.amazon.com/es/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/>
- [23] Web consultada per poder desplegar l'aplicació mitjançant travis [cons. el 10/03/2019]
<https://docs.travis-ci.com/user/deployment/s3/>
- [24] Panell de l'Sprint 2 del Taiga [cons. en múltiples ocasions]
<https://tree.taiga.io/project/carlotacatot-tfg-coink/taskboard/sprint-2-7904>
- [25] Web on es troben tots els components d'AntDesign [cons. en múltiples ocasions]
<https://ant.design/docs/react/introduce>
- [26] Documentació d'Amazon Web services per tal d'utilitzar els recursos de cognito mitjançant un SDK de JavaScript [cons. 13/03/2019]
https://docs.aws.amazon.com/es_es/cognito/latest/developerguide/using-amazon-cognito-user-identity-pools-javascript-examples.html#using-amazon-cognito-identity-user-pools-javascript-example-registering-user
- [27] Panell de l'Sprint 3 del Taiga [cons. en múltiples ocasions]
<https://tree.taiga.io/project/carlotacatot-tfg-coink/taskboard/sprint-3-5344>
- [28] Web on es troba tota la informació de les gràfiques utilitzades al sistema [cons. 05/04/2019]
<http://recharts.org/en-US/>
- [29] Blog on s'explica el procés a seguir per penjar el codi en Go a les funcions lambda [cons. en 30/03/2019]
<https://forums.aws.amazon.com/message.jspa?messageID=825738>
- [30] Blog on s'explica el procés a seguir per gestionar les bases de dades de dynamoDB amb Go [cons. en 26/03/2019]
<https://docs.aws.amazon.com/sdk-for-go/v1/developer-guide/using-dynamodb-with-go-sdk.html>

- [31] Blog on s'explica el procés a seguir per utilitzar els paràmetres que rep l'API Gateway a les funcions lambda [cons. en 28/03/2019]
<https://www.alexedwards.net/blog/serverless-api-with-go-and-aws-lambda>
- [32] Editor web per realitzar el disseny dels End-Points [cons. en múltiples ocasions]
<https://swagger.io/>
- [33] Documentació dels End-Points realitzats al primer sprint [cons. en múltiples ocasions]
<https://app.swaggerhub.com/apis-docs/carlotacb/Coink-API/1.0.1#/>
- [34] Documentació d'AWS per fer utilitzar el sistema de CDN de CloudFront [cons. 08/04/2019]
<https://aws.amazon.com/es/premiumsupport/knowledge-center/cloudfront-serve-static-website/>
- [35] Panell de l'Sprint 4 del Taiga [cons. en múltiples ocasions]
<https://tree.taiga.io/project/carlotacatot-tfg-coink/taskboard/sprint-4-3588>
- [36] Web d'on s'han extret les frases inspiradores de la pantalla principal [cons. 20/04/2019]
<https://www.coinc.es/blog/noticia/20-frases-celebres-y-refranes-ahorro>
- [37] Documentació dels End-Points realitzats al primer i segon sprint [cons. en múltiples ocasions]
<https://app.swaggerhub.com/apis-docs/carlotacb/Coink-API/1.0.2#/>
- [38] Panell de l'Sprint 5 del Taiga [cons. en múltiples ocasions]
<https://tree.taiga.io/project/carlotacatot-tfg-coink/taskboard/sprint-5-2398>
- [39] Documentació dels End-Points realitzats al primer, segon i tercer sprint [cons. en múltiples ocasions]
<https://app.swaggerhub.com/apis-docs/carlotacb/Coink-API/1.0.3#/>
- [40] Panell de l'Sprint 6 del Taiga [cons. en múltiples ocasions]
<https://tree.taiga.io/project/carlotacatot-tfg-coink/taskboard/sprint-6-1675>
- [41] Blog on s'explica la lògica de l'export utilitzat al sistema [cons. 10/05/2019]
<http://www.programmersought.com/article/3386496672/>
- [42] Sistema multiidioma utilitzat al sistema [cons. 08/05/2019]
<https://react.i18next.com/legacy-v9/step-by-step-guide#1-adding-more-languages>
- [43] Panell de l'Sprint 7 del Taiga [cons. en múltiples ocasions]
<https://tree.taiga.io/project/carlotacatot-tfg-coink/taskboard/sprint-7-1259>
- [44] Pàgina web anomenada Coink [última cons. 23/06/2019]
<http://coink.com/>
- [45] Documentació d'AWS utilitzada durant tot el desenvolupament [cons. en múltiples ocasions]
https://docs.aws.amazon.com/es_es/
- [46] StackOverflow - utilitzat per resoldre dubtes durant el desenvolupament [cons. en múltiples ocasions]
<https://stackoverflow.com/>
- [47] Web application architecture: principles, protocols, and practices - Shklar, L; Rosen, R, John Wiley, 2009
- [48] Mastering the requirements process: getting requirements right - Robertson, S.; Robertson, J, Addison-Wesley, 2006, 2013

- [49] Requirements engineering: fundamentals, principles, and techniques - Pohl, K, Springer, 2010.
- [50] Agile estimating and planning - Cohn, M, Prentice Hall Professional Technical Reference, 2006.
- [51] User stories applied: for agile software development - Cohn, M, Addison-Wesley , 2004.

Índex de figures

1.1	Taxa d'estalvi a les llars (percentatge de renda brut disponible) [2]	5
7.1	Diagrama de Gantt realitzat amb TeamGantt [11]	26
8.1	Diagrama de Gantt final realitzat amb TeamGantt [11]	30
13.1	Diagrama de les tecnologies utilitzades per la realització del sistema [13]	52
13.2	Diagrama de classes utilitzat dins el sistema [13]	53
13.3	Exemple del patró constructor	54
13.4	Exemple simple del patró observador	54
13.5	Exemple amb el cas de variables del patró Prototip	55
13.6	Exemple simple del patró fatxada	55
13.7	Exemple simple d'ús del patró factoria	56
13.8	Pantalla inicial de l'aplicació - login	58
13.9	Menú superior de l'aplicació	58
13.10	Primera pantalla del procés de registre	59
13.11	Exemple de email de verificació de compte	60
13.12	Segona pantalla del procés de registre	60
13.13	Tercera pantalla del procés de registre	61
13.14	Quarta pantalla del procés de registre	61
13.15	Part de la pantalla principal de l'aplicació	62
13.16	Exemple d'una de les frases inspiradores [36]	62
13.17	Pantalla on apareixen totes les transaccions de l'usuari	63
13.18	Modal per afegir una entrada	63
13.19	Modal per afegir una sortida	63
13.20	Fragment de la taula de transaccions on es veu una transacció editable	64
13.21	Pantalla on es veu una transacció anant a ser esborrada	64
13.22	Template de l'excel que s'exporta al premer el botó de exportar	64
13.23	Part de la pantalla que mostra les transaccions per any	65
13.24	Part superior de la pantalla que mostra les transaccions per any	65
13.25	Gràfiques que apareixen a la part inferior de la pantalla de transaccions per any i de la pantalla principal	66
13.26	Pantalla de contactes de l'usuari	66
13.27	Modal que permet afegir un contacte	67
13.28	Pantalla de grups de l'usuari	67
13.29	Modal que permet crear un nou grup	68
13.30	Pantalla d'informació d'un grup - Tots els moviments del grup	68
13.31	Pantalla d'informació d'un grup - Resum participants	69
13.32	Pantalla d'informació d'un grup - Editar grup	69
13.33	Pantalla d'informació de l'usuari	70
13.34	Modal que permet canviar la contrasenya de l'usuari	70
14.1	Backlog inicial - abans del Sprint 1	73
14.2	Panell de tasques de l'sprint 1	73
14.3	Backlog després de l'sprint 1	75
14.4	Panell de tasques de l'sprint 2	75
14.5	Backlog després de l'sprint 2	79

14.6	Panell de tasques de l'sprint 3	79
14.7	Endpoints dissenyats per l'Sprint 3	80
14.8	Backlog després de l'sprint 3	83
14.9	Panell de tasques de l'sprint 4	83
14.10	Endpoints dissenyats per l'Sprint 4	84
14.11	Backlog després de l'sprint 4	88
14.12	Panell de tasques de l'sprint 5	88
14.13	Endpoints dissenyats per l'Sprint 5	89
14.14	Backlog després de l'sprint 5	91
14.15	Panell de tasques de l'sprint 6	91
14.16	Backlog després de l'sprint 6	92
14.17	Panell de tasques de l'sprint 7	92
14.18	Configuració final API Gateway	94
14.19	Resum de les funcions que hi ha a les lambdes	94
14.20	Exemple de configuració d'una funció lambda programada en Go	95
14.21	Exemple de configuració d'una funció lambda programada en JavaScript	95
14.22	Configuració final DynamoDB	96
14.23	Configuració final Cognito	96
14.24	Gràfica de l'evolució dels punts d'histories d'usuari	97
14.25	Gràfica de la velocitat de realització dels punts d'histories d'usuari	97
15.1	Eines utilitzades pel desenvolupament del Front-End, AntDesign + React.js	99
15.2	Llenguatges utilitzats pel desenvolupament del Back-End, Golang i NodeJS	100
16.1	Exemple de testing per l'endpoint de GET /transactions	101
16.2	Gràfica de gastos en la versió dolenta	102
16.3	Fragment on es veu l'error dels e-mails	102
16.4	Fragment on es veu que es pot afegir un balanç que no sigui numèric	103
16.5	Fragment on es veu el color clar de la frase	103
16.6	Fragment on es veuen les categories per les transaccions Income	104
16.7	Fragment on es veuen que es poden afegir dates futures	104
17.1	Pàgina web del dia que es va descobrir [44]	105

Índex de taules

4.1	Taula comparativa de les funcionalitats de les aplicacions a l'estudi de mercat	15
6.1	NOT list inicial del projecte	19
6.2	NOT list final del projecte	20
7.1	Estimació inicial de temps	25
8.1	Estimació inicial de temps	29
9.1	Cost bloc <i>tasques inicials (GEP)</i>	31
9.2	Cost bloc <i>desenvolupament Back-End</i>	32
9.3	Cost bloc <i>desenvolupament Front-End</i>	32
9.4	Cost bloc <i>millores</i>	33
9.5	Cost bloc <i>documentació</i>	33
9.6	Taula dels costos directes	33
9.7	Taula dels costos indirectes	34
9.8	Taula resum dels costos del projecte	34
10.1	Cost reals bloc <i>tasques inicials (GEP)</i>	37
10.2	Cost real bloc <i>desenvolupament Back-End</i>	37
10.3	Cost real bloc <i>desenvolupament Front-End</i>	38
10.4	Cost real bloc <i>millores</i>	38
10.5	Cost real bloc <i>documentació</i>	38
10.6	Taula dels costos directes	39
10.7	Taula resum dels costos del projecte	39
11.1	Matriu de sostenibilitat	41
14.1	Hores invertides al desenvolupament Back-End durant l'Sprint 1	72
14.2	Hores invertides al desenvolupament Front-End durant l'Sprint 1	72
14.3	Hores invertides al desenvolupament Back-End durant l'Sprint 2	74
14.4	Hores invertides al desenvolupament Front-End durant l'Sprint 2	75
14.5	Hores invertides al desenvolupament Back-End durant l'Sprint 3	78
14.6	Hores invertides al desenvolupament Front-End durant l'Sprint 3	78
14.7	Hores invertides al desenvolupament Back-End durant l'Sprint 4	82
14.8	Hores invertides al desenvolupament Front-End durant l'Sprint 4	82
14.9	Hores invertides al desenvolupament Back-End durant l'Sprint 5	87
14.10	Hores invertides al desenvolupament Front-End durant l'Sprint 5	87
14.11	Hores invertides al desenvolupament Front-End durant l'Sprint 6	90
14.12	Hores invertides al desenvolupament Front-End durant l'Sprint 7	92
14.13	Hores invertides en millores durant l'Sprint 7	92
14.14	Resum d'hores invertides en el desenvolupament de Back-End	93
14.15	Resum d'hores invertides en el desenvolupament de Front-End	93
14.16	Resum d'hores invertides en millores	93